CAP 4453
Robot Vision
Dr. Gonzalo Vaca-Castaño
gonzalo.vacacastano@ucf.edu
Administrative details

• Correction of the midterm exam
Credits

• Some slides comes directly from:
  • Yosesh Rawat
  • Andrew Ng
Robot Vision

17. Introduction to Deep Learning II
Outline

• Fully connected Neural network
  • Activation functions:
    • Forward and backward
  • Back propagation
  • Network definitions
  • Initialization
  • Training
    • Hyper parameters
    • Gradient updates: RMS prop,
    • Amount of training data
    • Batch normalization
• Dataset
  • Train set, test set, validation set
  • Bias and variance

• Implementation network to solve digit identification
Fully connected networks: The math
A REVIEW

Fully connected Neural network

- A deep network is a neural network with many layers
- A neuron in a linear function followed for an activation function
- Activation function must be non-linear
- A loss function measures how close is the created function (network) from a desired output
- The “training” is the process of find parameters (‘weights’) that reduces the loss functions
- Updating the weights as \( w_{\text{new}} = w_{\text{prev}} - \alpha \frac{dJ}{dW} \) reduces the loss
- An algorithm named back-propagation allows to compute \( \frac{dJ}{dW} \) for all the weights of the network in 2 steps: 1 forward, 1 backward
A Neuron
A REVIEW

\[ z = w^T x \quad y = f(z) = f(w^T x) \]

\[ x = [x_1, x_2, x_3, 1] \]

Activations and their derivatives

SIGMOID

\[ f(z) = \frac{1}{1 + \exp(-z)} \]

LOGISTIC FUNCTION

\[ f'(z) = f(z)(1 - f(z)) \]

\[ f(x) = \frac{1}{1 + e^{-x}} \]

\[ \frac{d}{dz} f(x) = \frac{e^x \cdot (1 + e^x) - e^x \cdot e^x}{(1 + e^x)^2} = \frac{e^x}{(1 + e^x)^2} = f(x)(1 - f(x)) \]
A Neuron
A REVIEW

\[ z = w^T x \]
\[ y = f(z) = f(w^T x) \]
\[ x = [x_1, x_2, x_3, 1] \]

Activations and their derivatives

**SIGMOID**
\[ f(z) = \frac{1}{1 + \exp(-z)} \]
\[ f'(z) = f(z)(1 - f(z)) \]

**LOGISTIC FUNCTION**
\[ f(z) = \tanh(z) \]
\[ f'(z) = (1 - f^2(z)) \]

**SOFTPLUS or SMOOTHRELU FUNCTION**
\[ f(z) = \log(1 + \exp(z)) \]
\[ f'(z) = \frac{1}{1 + \exp(-z)} \]
How to minimize a function?

\[ w_{\text{new}} = w_{\text{prev}} - \alpha \frac{dJ}{dW} \]

IN OUR CASE THE LOSS FUNCTION

Repeat until there is almost no change

HOW TO COMPUTE THIS GRADIENT?
Gradient descent

\[ f(x) \]
General approach

Pick random starting point.
General approach

Compute gradient at point (analytically or by finite differences)
General approach

Move along parameter space in direction of negative gradient

\[ a_2 = a_1 - \gamma \nabla f(a_1) \]

\( \gamma = \) amount to move
   = learning rate
General approach

Move along parameter space in direction of negative gradient.

\[ a_3 = a_2 - \gamma \nabla f(a_2) \]

\( \gamma \) = amount to move
\( = \) learning rate
General approach

Stop when we don’t move any more.

\[ a_{n-1} - \gamma \nabla f(a_{n-1}) = 0 \]
Gradient Descent

• The gradient is the direction of fastest increase in $J(X)$

• Updating the weights as $w_{new} = w_{prev} - \alpha \frac{dJ}{dW}$ reduces the loss

**The Approach of Gradient Descent**

• Iterative solution:
  - Start at some point
  - Find direction in which to shift this point to decrease error
    • This can be found from the derivative of the function
      - A positive derivative $\Rightarrow$ moving left decreases error
      - A negative derivative $\Rightarrow$ moving right decreases error
  - Shift point in this direction

**Overall Gradient Descent Algorithm**

• Initialize:
  - $x^0$
  - $k = 0$

• While $|f(x^{k+1}) - f(x^k)| > \varepsilon$
  - $x^{k+1} = x^k - \eta^k \nabla f(x^k)^T$
  - $k = k + 1$
Train with Gradient Descent

- $x^i, y^i = n$ training examples
- $f(x) =$ feed forward network
- $L(x, y; \theta) =$ some *loss function*

*Loss function* measures how ‘good’ our network is at classifying the training examples wrt. the parameters of the model (the perceptron weights).
Loss Function

• Way to define how good the network is performing
  • In terms of prediction
• Network training (Optimization)
  • Find the best network parameters to minimize the loss

\[
\text{Total Error} (W) = \frac{1}{N} \sum_{i=1}^{N} L_i(f(x_i, W), y_i)
\]
Loss Functions and total Error

Cross-Entropy
a.k.a. log loss

\[
\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{n} t_j \log(p_j)
\]

\(t_j\) is the truth label
\(p_j\) is the Softmax probability for the \(j^{th}\) class

N samples

Binary Cross Entropy

\[
-\frac{1}{N} \sum_{i=1}^{N} \left( y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right)
\]

Ground-truth

Predicted value

• Mean squared error (MSE)

\[
\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2
\]
\[ L_{CE} = - \sum_{i=1}^{n} t_i \log(p_i), \] for \( n \) classes,

where \( t_i \) is the truth label and \( p_i \) is the Softmax probability for the \( i^{th} \) class.
**Softmax**

\[ \sigma(\tilde{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \]

Used to interpret outputs as probabilities

\[
\begin{bmatrix}
P(\text{cat}) \\
P(\text{dog})
\end{bmatrix}
= \sigma(\begin{bmatrix}
1.2 \\
0.3
\end{bmatrix})
\]

\[
= \begin{bmatrix}
\frac{e^{1.2}}{e^{1.2} + e^{0.3}} \\
\frac{e^{0.3}}{e^{1.2} + e^{0.3}}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
0.71 \\
0.29
\end{bmatrix}
\]

<table>
<thead>
<tr>
<th>( \tilde{z} )</th>
<th>The input vector to the softmax function, made up of ((z_0, \ldots, z_K))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z_i )</td>
<td>All the ( z ) values are the elements of the input vector to the softmax function, and they can take any real value, positive, zero or negative. For example a neural network could have output a vector such as ((-0.62, 0.12, 2.63)), which is not a valid probability distribution, hence why the softmax would be necessary.</td>
</tr>
<tr>
<td>( e^{z_i} )</td>
<td>The standard exponential function is applied to each element of the input vector. This gives a positive value above 0, which will be very small if the input was negative, and very large if the input was large. However, it is still not fixed in the range ((0, 1)) which is what is required of a probability.</td>
</tr>
<tr>
<td>( \sum_{j=1}^{K} e^{z_j} )</td>
<td>The term on the bottom of the formula is the normalization term. It ensures that all the output values of the function will sum to 1 and each be in the range ((0, 1)), thus constituting a valid probability distribution.</td>
</tr>
<tr>
<td>( K )</td>
<td>The number of classes in the multi-class classifier.</td>
</tr>
</tbody>
</table>
Learning rate

\[ w_{\text{new}} = w_{\text{prev}} - \alpha \frac{dJ}{dW} \]
Notation

- The input layer is the 0th layer
- We will represent the output of the i-th perceptron of the kth layer as \( y_i^{(k)} \)
  - Input to network: \( y_i^{(0)} = x_i \)
  - Output of network: \( y_i = y_i^{(N)} \)
- We will represent the weight of the connection between the i-th unit of the k-1th layer and the jth unit of the k-th layer as \( w_{ij}^{(k)} \)
  - The bias to the jth unit of the k-th layer is \( b_j^{(k)} \)
Training steps

• Define network
• Loss function
• Initialize network parameters
• Get training data
  • Prepare batches
• Feedforward one batch
  • Compute loss
  • Update network parameters
  • Repeat
How to minimize a function?

In our case the loss function $w_{new} = w_{prev} - \alpha \frac{dJ}{dW}$

Repeat until there is almost not change

\[ w_{new} = w_{prev} - \alpha \frac{dJ}{dW} \]
Training Neural Nets through Gradient Descent

Total training error:

\[ Err = \frac{1}{T} \sum_t Div(Y_t, d_t) \]

- Gradient descent algorithm:
  - Initialize all weights and biases \( \{w_{ij}^{(k)}\} \)
    - Using the extended notation: the bias is also a weight
  - Do:
    - For every layer \( k \) for all \( i, j \), update:
      - \( w_{ij}^{(k)} = w_{ij}^{(k)} - \eta \frac{dErr}{dw_{ij}^{(k)}} \)
  - Until \( Err \) has converged

Example: L2

\[ Div = \frac{1}{2} (y_t - d_t)^2 \]

\[ \frac{dDiv}{dy_t} = (y_t - d_t) \]
The derivative

Total training error:

$$Err = \frac{1}{T} \sum_t Div(Y_t, d_t)$$

• Computing the derivative

Total derivative:

$$\frac{dErr}{d\omega_{i,j}^{(k)}} = \frac{1}{T} \sum_t \frac{dDiv(Y_t, d_t)}{d\omega_{i,j}^{(k)}}$$
The derivative

Total training error:

$$Err = \frac{1}{T} \sum_t Div(Y_t, d_t)$$

Total derivative:

$$\frac{dErr}{dw^{(k)}_{i,j}} = \frac{1}{T} \sum_t \frac{dDiv(Y_t, d_t)}{dw^{(k)}_{i,j}}$$

- So we must first figure out how to compute the derivative of divergences of individual training inputs
Calculus Refresher: Basic rules of calculus

For any differentiable function
\[ y = f(x) \]
with derivative
\[ \frac{dy}{dx} \]
the following must hold for sufficiently small \( \Delta x \)
\[ \Delta y \approx \frac{dy}{dx} \Delta x \]

For any differentiable function
\[ y = f(x_1, x_2, \ldots, x_M) \]
with partial derivatives
\[ \frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \ldots, \frac{\partial y}{\partial x_M} \]
the following must hold for sufficiently small \( \Delta x_1, \Delta x_2, \ldots, \Delta x_M \)
\[ \Delta y \approx \frac{\partial y}{\partial x_1} \Delta x_1 + \frac{\partial y}{\partial x_2} \Delta x_2 + \ldots + \frac{\partial y}{\partial x_M} \Delta x_M \]
Calculus Refresher: Chain rule

For any nested function \( y = f(g(x)) \)

\[
\frac{dy}{dx} = \frac{\partial y}{\partial g(x)} \frac{dg(x)}{dx}
\]

Check - we can confirm that:

\[
\Delta y = \frac{dy}{dx} \Delta x
\]

\( z = g(x) \) → \( \Delta z = \frac{dg(x)}{dx} \Delta x \)

\( y = f(z) \) → \( \Delta y = \frac{dy}{dz} \Delta z = \frac{dy}{dz} \frac{dg(x)}{dx} \Delta x \)
Calculus Refresher: Distributed Chain rule

\[ y = f(g_1(x), g_1(x), \ldots, g_M(x)) \]

\[
\frac{dy}{dx} = \frac{\partial y}{\partial g_1(x)} \frac{dg_1(x)}{dx} + \frac{\partial y}{\partial g_2(x)} \frac{dg_2(x)}{dx} + \cdots + \frac{\partial y}{\partial g_M(x)} \frac{dg_M(x)}{dx}
\]

Check: \[ \Delta y = \frac{dy}{dx} \Delta x \]

\[
\Delta y = \frac{\partial y}{\partial g_1(x)} \Delta g_1(x) + \frac{\partial y}{\partial g_2(x)} \Delta g_2(x) + \cdots + \frac{\partial y}{\partial g_M(x)} \Delta g_M(x)
\]

\[
\Delta y = \frac{\partial y}{\partial g_1(x)} \frac{dg_1(x)}{dx} \Delta x + \frac{\partial y}{\partial g_2(x)} \frac{dg_2(x)}{dx} \Delta x + \cdots + \frac{\partial y}{\partial g_M(x)} \frac{dg_M(x)}{dx} \Delta x
\]

\[
\Delta y = \left( \frac{\partial y}{\partial g_1(x)} \frac{dg_1(x)}{dx} + \frac{\partial y}{\partial g_2(x)} \frac{dg_2(x)}{dx} + \cdots + \frac{\partial y}{\partial g_M(x)} \frac{dg_M(x)}{dx} \right) \Delta x
\]
Distributed Chain Rule: Influence Diagram

- Small perturbations in $x$ cause small perturbations in each of $g_1 \ldots g_M$, each of which individually additively perturbs $y$
A first closer look at the network

- Showing a tiny 2-input network for illustration
  - Actual network would have many more neurons and inputs
- Expanded with all weights and activations shown
- The overall function is differentiable w.r.t every weight, bias and input
**Forward Computation**

\[ y_i^{(0)} = x_i \]

**ITERATE FOR** \( k = 1:N \)

**for** \( j = 1:\text{layer-width} \)

\[ z_j^{(k)} = \sum_i w_{ij}^{(k)} y_i^{(k-1)} \]

\[ y_j^{(k)} = f_k \left( z_j^{(k)} \right) \]
Gradients: Backward Computation

\[ \text{Div} = \frac{1}{2} (y_t - d_t)^2 \]

\[ \frac{d\text{Div}}{dy_i} = (y_t - d_t) \]
Gradients: Backward Computation

\[
\frac{\partial \text{Div}(Y, d)}{\partial y_i} = \frac{\partial \text{Div}(Y, d)}{\partial y_i^{(N)}}
\]

\[
\frac{\partial \text{Div}}{\partial z_i^{(N)}} = \frac{\partial y_i^{(N)}}{\partial y_i} \frac{\partial \text{Div}}{\partial y_i} = f_N'(z_i^{(N)}) \frac{\partial \text{Div}}{\partial y_i^{(N)}}
\]
\[ y_i^{[N]} = f(z_i^{[N]}) \]
\[ \frac{\partial y_i^{[N]}}{\partial z_i^{[N]}} = f^{[N]'}(z_i^{[N]}) \]

\[ \frac{\partial \text{Div}(Y, d)}{\partial y_i} = \frac{\partial \text{Div}(Y, d)}{\partial y_i^{(N)}} \]

\[ \frac{\partial \text{Div}}{\partial z_i^{(N)}} = \frac{\partial y_i^{(N)}}{\partial z_i^{(N)}} \frac{\partial \text{Div}}{\partial y_i} = f_N^{'}(z_i^{(N)}) \frac{\partial \text{Div}}{\partial y_i^{(N)}} \]
Gradients: Backward Computation

\[ \frac{\partial \text{Div}(Y, d)}{\partial y_i^{(N)}} = \frac{\partial \text{Div}(Y, d)}{\partial Y_i} \]

\[ \frac{\partial f_N}{\partial z_i^{(N)}} = f_N'(z_i^{(N)}) \frac{\partial \text{Div}}{\partial y_i^{(N)}} \]

\[ z_i^{(N)} \text{ computed during the forward pass} \]
Gradients: Backward Computation

\[ \frac{\partial \text{Div}(Y, d)}{\partial Y_i} = \frac{\partial \text{Div}(Y, d)}{\partial y_i^{(N)}} \]

Derivative of the activation function of Nth layer

\[ \frac{\partial \text{Div}}{\partial z_i^{(N)}} = \frac{\partial y_i^{(N)}}{\partial z_i^{(N)}} \frac{\partial \text{Div}}{\partial Y_i} = f_N'(z_i^{(N)}) \frac{\partial \text{Div}}{\partial y_i^{(N)}} \]
Gradients: Backward Computation

\[ z_j^{[N]} = w^T y_i^{[N-1]} \]

\[
\frac{\partial \text{Div}(Y, d)}{\partial Y_i} = \frac{\partial \text{Div}(Y, d)}{\partial y_i^{(N)}}
\]

\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = \frac{\partial z_j^{(k)}}{\partial w_{ij}^{(k)}} \frac{\partial \text{Div}}{\partial z_j^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial z_i^{(N)}} = f_N(z_i^{(N)}) \frac{\partial \text{Div}}{\partial y_i^{(N)}}
\]
Gradients: Backward Computation

\[
\frac{\partial \text{Div}}{\partial y_i^{(N-1)}} = \sum_j \frac{\partial z_j^{(N)}}{\partial y_i^{(N-1)}} \frac{\partial \text{Div}}{\partial z_j^{(N)}} = \sum_j w_{ij}^{(N)} \frac{\partial \text{Div}}{\partial z_j^{(N)}}
\]

Because:

\[
\frac{\partial z_j^{(N)}}{\partial y_i^{(N-1)}} = w_{ij}^{(N)}
\]

\[
\frac{\partial \text{Div}}{\partial y_i^{(N)}} = \frac{\partial \text{Div}(Y, d)}{\partial y_i^{(N)}}
\]

\[
\frac{\partial \text{Div}}{\partial z_i^{(N)}} = f'_N(z_i^{(N)}) \frac{\partial \text{Div}}{\partial y_i^{(N)}}
\]
Gradients: Backward Computation

\[
\frac{\partial \text{Div}}{\partial y_i^{(N-1)}} = \sum_j \frac{\partial z_j^{(N)}}{\partial y_i^{(N-1)}} \frac{\partial \text{Div}}{\partial z_j^{(N)}} = \sum_j w_{ij}^{(N)} \frac{\partial \text{Div}}{\partial z_j^{(N)}}
\]

Because:

\[
\frac{\partial z_j^{(N)}}{\partial y_i^{(N-1)}} = w_{ij}^{(N)}
\]

\[
z_j^{[N]} = w^T y_i^{[N-1]}
\]

But in this case the input is the output from previous layer.
**Gradients: Backward Computation**

\[
\frac{\partial \text{Div}}{\partial z_i^{(N)}} = f'_N \left( z_i^{(N)} \right) \frac{\partial \text{Div}}{\partial y_i^{(N)}}
\]

\[
\frac{\partial \text{Div}}{\partial y_i^{(N-1)}} = \sum_j w_{ij}^{(N)} \frac{\partial \text{Div}}{\partial z_j^{(N)}}
\]

\[
\frac{\partial \text{Div}}{\partial y_i^{(k)}} = f'_k \left( z_i^{(k)} \right) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]

**Computed during the forward pass**

- **Div(Y,d)**
- **\( f_N \)**
- **\( y^{(N)} \)**
- **\( z^{(N)} \)**
- **\( y^{(k)} \)**
- **\( z^{(k)} \)**
- **\( y^{(k-1)} \)**
- **\( z^{(k-1)} \)**
- **\( y^{(N-1)} \)**
- **\( z^{(N-1)} f_{N-1} \)**
Gradients: Backward Computation

\[
\frac{\partial \text{Div}}{\partial y_{i}^{(k-1)}} = \sum_{j} \frac{\partial z_{j}^{(k)}}{\partial y_{i}^{(k-1)}} \frac{\partial \text{Div}}{\partial z_{j}^{(k)}} = \sum_{j} w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_{j}^{(k)}}
\]
$z_j^{[N]} = w^T y_i^{[N-1]}$

$\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \frac{\partial y_i^{(k-1)}}{\partial z_j^{(k)}} \frac{\partial \text{Div}}{\partial z_j^{(k)}}$

$\frac{\partial \text{Div}}{\partial y_i^{(k)}} = \frac{\partial \text{Div}(Y, d)}{\partial y_i^{(N)}}$

$\frac{\partial \text{Div}}{\partial z_i^{(N)}} = f'_N(z_i^{(N)}) \frac{\partial \text{Div}}{\partial y_i^{(N)}}$
Gradients: Backward Computation

For $k = N-1$
For $i = 1$ to $layer - width$

\[
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f'_k(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

Initialize: Gradient w.r.t. network output

\[
\frac{\partial \text{Div}}{\partial y_i} = \frac{\partial \text{Div}(Y, d)}{\partial y_i^{(N)}}
\]
Training by BackProp

- Initialize all weights \((W^{(1)}, W^{(2)}, \ldots, W^{(K)})\)
- Do:
  - Initialize \(Err = 0\); For all \(i, j, k\), initialize \(\frac{dErr}{dw_{i,j}^{(k)}} = 0\)
  - For all \(t = 1:T\) (Loop over training instances)
    - **Forward pass:** Compute
      - Output \(Y_t\)
      - \(Err += Div(Y_t, d_t)\)
    - **Backward pass:** For all \(i, j, k\):
      - Compute \(\frac{dDiv(Y_t, d_t)}{dw_{i,j}^{(k)}}\)
      - Compute \(\frac{dErr}{dw_{i,j}^{(k)}} += \frac{dDiv(Y_t, d_t)}{dw_{i,j}^{(k)}}\)
    - For all \(i, j, k\), update:
      \[
      w_{i,j}^{(k)} = w_{i,j}^{(k)} - \frac{\eta}{T} \frac{dErr}{dw_{i,j}^{(k)}}
      \]
- Until \(Err\) has converged
Exercise

\[ Div = \frac{(\hat{y}_i - y_i)^2}{2} \]

Expected output: 1

\[ (y_i - \hat{y}_i) \]

Activations and their derivatives

<table>
<thead>
<tr>
<th>Activation</th>
<th>Formula</th>
<th>Derivative</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGMOID</td>
<td>( f(z) = \frac{1}{1 + \exp(-z)} )</td>
<td>( f'(z) = f(z)(1 - f(z)) )</td>
</tr>
<tr>
<td>LOGISTIC FUNCTION</td>
<td>( f(z) = \tanh(z) )</td>
<td>( f'(z) = (1 - f^2(z)) )</td>
</tr>
<tr>
<td>tanh</td>
<td>( f(z) = \begin{cases} 0, &amp; z \leq 0 \ z, &amp; z \geq 0 \end{cases} )</td>
<td>( f'(z) = \frac{1}{1 + \exp(-z)} )</td>
</tr>
</tbody>
</table>

Softplus or SmoothReLU function

Gradients: Backward Computation

Initialize: Gradient w.r.t network output

For \( K = N-1 \)

\[ \frac{\partial Div}{\partial y_i} = \frac{\partial Div(Y, d)}{\partial y_i} = \frac{\partial}{\partial y_i} f_i(N) \frac{\partial Div}{\partial f_i(N)} \]

\[ \frac{\partial Div}{\partial z_{ij}} = \sum w_{ij} \frac{\partial Div}{\partial y_i} \]
Example: Forward

\[ z_1^{[1]} = w_{11}^{[1]} x_1 + w_{21}^{[1]} x_2 \]
\[ = 0.3 \times 1.1 + 0.4 \times 0.4 \]
\[ = 0.17 \]

\[ z_2^{[1]} = w_{12}^{[1]} x_1 + w_{22}^{[1]} x_2 \]
\[ = -0.3 \times 1.1 + 0.6 \times 0.4 \]
\[ = -0.019 \]

\[ y_1^{[1]} = \tanh(z_1^{[1]}) = \tanh(0.17) = 0.1683 \]
\[ y_2^{[1]} = \tanh(z_2^{[1]}) = \tanh(-0.019) = -0.0897 \]
Example: Forward

\[ y_1^{[1]} = 0.1683 \]
\[ y_2^{[1]} = -0.0897 \]

\[
\begin{align*}
z_1^{[2]} &= w_{11}^{[2]} y_1^{[1]} + w_{21}^{[2]} y_2^{[1]} \\
&= 0.9 * 0.1683 - 0.3 * 0.0897 \\
&= 0.124615
\end{align*}
\]

Expected output: 1

\[
Div = \frac{(\hat{y}_i - y_i)^2}{2}
\]

**LAYER 2**

\[
y_1^{[2]} = \text{logistic} \left( z_1^{[2]} \right) = \frac{1}{1 + e^{z_1^{[2]}}} = \frac{1}{1 + e^{0.1246}} = 0.531113
\]
Example: Backward

\[
\begin{align*}
  z_1^{[1]} &= 0.17 \quad y_1^{[1]} = 0.1683 \\
  z_2^{[1]} &= -0.019 \quad y_2^{[1]} = -0.0897 \\
  z_1^{[2]} &= 0.1246 \quad y_1^{[2]} = 0.5311
\end{align*}
\]

\[
\text{Expected output: 1}
\]

\[
\text{Div} = \frac{(\hat{y}_i - y_i)^2}{2}
\]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = (y_1^{[2]} - \hat{y})
\]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = (0.5311 - 1)
\]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688
\]

\[
\text{initialize} \quad z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \\
\text{initialize} \quad z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \\
\text{initialize} \quad z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311
\]
Example: Backward

\[
\begin{align*}
z_1^{[1]} &= 0.17 \quad y_1^{[1]} = 0.1683 \\
z_2^{[1]} &= -0.019 \quad y_2^{[1]} = -0.0897 \\
z_1^{[2]} &= 0.1246 \quad y_1^{[2]} = 0.5311 \\
\end{align*}
\]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688
\]

For \( k = N-1 \)
For \( i = 1: \text{layer} - \text{width} \)

\[
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f'_k \left( z_i^{(k)} \right) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

Expected output: 1

\[K=2\]
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[ \frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688 \]

For \( k = N \), 1
For \( i = 1: \text{layer-width} \)
\[ \frac{\partial \text{Div}}{\partial z_i^{(k)}} = f_k' \left( z_i^{(k)} \right) \frac{\partial \text{Div}}{\partial y_i^{(k)}} \]
\[ \frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]
\[ \frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]

\[ f(z) = \frac{1}{1 + \exp(-z)} \quad \text{LOGISTIC FUNCTION} \quad f'(z) = f(z)(1 - f(z)) \]

\[ K=2 \]

Expected output: 1
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[ \frac{\partial D_{\text{div}}}{\partial y_1^{[2]}} = -0.4688 \]

\[ \frac{\partial D_{\text{div}}}{\partial z_1^{[2]}} = f_2'(z_1^{[2]}) \frac{\partial D_{\text{div}}}{\partial y_1^{[2]}} \]

\[ \frac{\partial D_{\text{div}}}{\partial z_2^{[2]}} = f_2(z_1^{[2]}) \left(1 - f_2(z_1^{[2]})\right) \frac{\partial D_{\text{div}}}{\partial y_1^{[2]}} \]

For \( k = N - 1 \)
For \( i = 1: \text{layer - width} \)

\[ \frac{\partial D_{\text{div}}}{\partial z_i^{(k)}} = f_k' \left(z_i^{(k)}\right) \frac{\partial D_{\text{div}}}{\partial y_i^{(k)}} \]

\[ \frac{\partial D_{\text{div}}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial D_{\text{div}}}{\partial z_j^{(k)}} \]

\[ \frac{\partial D_{\text{div}}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial D_{\text{div}}}{\partial z_j^{(k)}} \]

SIGMOID: \[ f(z) = \frac{1}{1 + \exp(-z)} \]

LOGISTIC FUNCTION: \[ f'(z) = f(z)(1 - f(z)) \]

Expected output: 1

Layer 2

\( K = 2 \)
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[ \frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688 \]

\[ \frac{\partial \text{Div}}{\partial y_1^{[2]}}, \quad \frac{\partial \text{Div}}{\partial y_2^{[2]}} \]

\[ \text{For } k = N-1 \]
\[ \text{For } i = 1: \text{layer} - \text{width} \]

\[ \frac{\partial \text{Div}}{\partial z_i^{(k)}} = f_k \left( z_i^{(k)} \right) \frac{\partial \text{Div}}{\partial y_i^{(k)}} \]

\[ \frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]

\[ \frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]

\[ \frac{\partial \text{Div}}{\partial z_1^{[2]}} = f_2 \left( z_1^{[2]} \right) \left( 1 - f_2 \left( z_1^{[2]} \right) \right) \frac{\partial \text{Div}}{\partial y_1^{[2]}} \]

\[ \frac{\partial \text{Div}}{\partial z_2^{[2]}} = y_1^{[2]} \left( 1 - y_1^{[2]} \right) \frac{\partial \text{Div}}{\partial y_1^{[2]}} \]

\[ \frac{\partial \text{Div}}{\partial z_1^{[2]}} = f_2' \left( y_1^{[2]} \right) \frac{\partial \text{Div}}{\partial y_1^{[2]}} \]

\[ \frac{\partial \text{Div}}{\partial z_2^{[2]}} = f_2' \left( y_2^{[2]} \right) \frac{\partial \text{Div}}{\partial y_2^{[2]}} \]

Expected output: 1

K = 2
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688
\]

For \( k = N \cdot 1 \)
For \( i = 1: \text{layer-width} \)

\[
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f_k'(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]
\[
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]
\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

Layer 2

\[
\frac{\partial \text{Div}}{\partial z_1^{[2]}} = f_2'(z_1^{[2]}) \frac{\partial \text{Div}}{\partial y_1^{[2]}}
\]
\[
\frac{\partial \text{Div}}{\partial z_2^{[2]}} = f_2(z_2^{[2]})(1 - f_2(z_2^{[2]})) \frac{\partial \text{Div}}{\partial y_1^{[2]}}
\]
\[
\frac{\partial \text{Div}}{\partial z_1^{[2]}} = y_1^{[2]}(1 - y_1^{[2]}) \frac{\partial \text{Div}}{\partial y_1^{[2]}}
\]
\[
\frac{\partial \text{Div}}{\partial z_2^{[2]}} = 0.5311(1 - 0.5311)(-0.4688)
\]

Expected output: 1
Example: Backward

\[
\begin{align*}
z_1^{[1]} &= 0.17, \quad y_1^{[1]} = 0.1683 \\
z_2^{[1]} &= -0.019, \quad y_2^{[1]} = -0.0897 \\
z_1^{[2]} &= 0.1246, \quad y_1^{[2]} = 0.5311
\end{align*}
\]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688
\]

For \( k = N \cdot 1 \)
For \( i = 1: \text{layer-width} \)

\[
\begin{align*}
\frac{\partial \text{Div}}{\partial z_i^{(k)}} &= f_k'(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}} \\
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} &= \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \\
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} &= y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\end{align*}
\]

Layer 2

\( K = 2 \)

Expected output: 1
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688 \\
\frac{\partial \text{Div}}{\partial y_1^{[1]}} = \left( -0.1167 \right) \\
\frac{\partial \text{Div}}{\partial z_1^{[2]}} = (-0.1167)
\]

For \( k = N-1 \)
For \( i = 1: \text{layer} - \text{width} \)
\[
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f'_k \left( z_i^{(k)} \right) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]
\[
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]
\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial y_1^{[1]}} = \sum_{j=1}^{1} w_{1j}^{[2]} \frac{\partial \text{Div}}{\partial z_j^{[2]}}
\]
\[ \frac{\partial \text{Div}}{\partial y_1^{[1]}} = w_{11}^{[2]} \frac{\partial \text{Div}}{\partial z_1^{[2]}} \]
\[ \frac{\partial \text{Div}}{\partial y_1^{[1]}} = 0.9(-0.1167) \]
\[
\frac{\partial \text{Div}}{\partial y_1^{[1]}} = -0.10509
\]
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

For \( k = N-1 \)
For \( i = 1 : \text{layer - width} \)
\[
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f'_k(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]
\[
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]
\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = 1 \sum_j w_{2j}^{[2]} \frac{\partial \text{Div}}{\partial z_j^{[2]}}
\]
\[
\frac{\partial \text{Div}}{\partial y_2^{[2]}} = w_{21}^{[2]} \frac{\partial \text{Div}}{\partial z_1^{[2]}}
\]
\[
\frac{\partial \text{Div}}{\partial y_2^{[1]}} = 0.3(-0.1167)
\]
\[
\frac{\partial \text{Div}}{\partial y_2^{[1]}} = -0.03503
\]

Expected output: 1

\[ x_1 \]
\[ x_2 \]
\[ \hat{y} \]

\[ \begin{align*}
\text{tanh} & : 0.3 \\
\text{logistic} & : 0.9
\end{align*} \]
Example: Backward

\[
\begin{align*}
z_1^{[1]} & = 0.17 \quad y_1^{[1]} = 0.1683 \\
z_2^{[1]} & = -0.019 \quad y_2^{[1]} = -0.0897 \\
z_1^{[2]} & = 0.1246 \quad y_1^{[2]} = 0.5311
\end{align*}
\]

\[
\begin{align*}
\frac{\partial \text{Div}}{\partial y_1^{[2]}} & = -0.4688 \\
\frac{\partial \text{Div}}{\partial z_1^{[2]}} & = (-0.1167) \\
\frac{\partial \text{Div}}{\partial y_1^{[1]}} & = -0.10509 \\
\frac{\partial \text{Div}}{\partial y_2^{[1]}} & = -0.03503
\end{align*}
\]

For \( k = N, 1 \)
For \( i = 1: \text{layer} - \text{width} \)
\[
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f_k'(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]
\[
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]
\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = y_1^{[1]} \frac{\partial \text{Div}}{\partial z_1^{[2]}}
\]
\[
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = y_1^{[1]} \frac{\partial \text{Div}}{\partial z_1^{[2]}} = 0.1683(-0.1167)
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = (-0.01966)
\]

Expected output: 1
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688 \\
\frac{\partial \text{Div}}{\partial z_2^{[2]}} = (-0.1167) \\
\frac{\partial \text{Div}}{\partial y_1^{[1]}} = -0.10509 \\
\frac{\partial \text{Div}}{\partial y_2^{[1]}} = -0.03503
\]

For \( k = N - 1 \)
For \( i = 1: \text{layer} - \text{width} \)

\[
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \\
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f'_k(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}} \\
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial w_{21}^{[2]}} = y_1^{[1]} \frac{\partial \text{Div}}{\partial z_1^{[2]}} \\
\frac{\partial \text{Div}}{\partial w_{21}^{[2]}} = (-0.0897)(-0.1167)
\]

\[
\frac{\partial \text{Div}}{\partial w_{21}^{[2]}} = 0.010481 \\
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = (-0.01966)
\]
Example: Backward

\[
\begin{align*}
z_1^{[1]} &= 0.17, \quad y_1^{[1]} = 0.1683 \\
z_2^{[1]} &= -0.019, \quad y_2^{[1]} = -0.0897 \\
z_1^{[2]} &= 0.1246, \quad y_1^{[2]} = 0.5311 \\
\end{align*}
\]

\[
\begin{align*}
\frac{\partial \text{Div}}{\partial y_1^{[2]}} &= -0.4688 \\
\frac{\partial \text{Div}}{\partial z_1^{[2]}} &= (-0.1167) \\
\frac{\partial \text{Div}}{\partial y_1^{[1]}} &= -0.10509 \\
\frac{\partial \text{Div}}{\partial y_2^{[1]}} &= -0.03503 \\
\end{align*}
\]

For \( k = N-1 \)

For \( i = 1: \text{layer} - \text{width} \)

\[
\begin{align*}
\frac{\partial \text{Div}}{\partial z_i^{[k]}} &= f_k'(z_i^{[k]}) \frac{\partial \text{Div}}{\partial y_i^{[k]}} \\
\frac{\partial \text{Div}}{\partial y_i^{[k-1]}} &= \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{[k]}} \\
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} &= y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{[k]}} \\
\end{align*}
\]

\[
\begin{align*}
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} &= (-0.01966) \\
\frac{\partial \text{Div}}{\partial w_{21}^{[2]}} &= 0.010481 \\
\end{align*}
\]

\[
\begin{align*}
f(x) &= \tanh(x) \\
f'(x) &= (1 - f^2(x))
\end{align*}
\]

Expected output: 1

\[
\begin{align*}
\frac{\partial \text{Div}}{\partial y_1^{[1]}} &= f_1'(z_1^{[1]}) \frac{\partial \text{Div}}{\partial y_1^{[1]}} \\
\frac{\partial \text{Div}}{\partial y_2^{[1]}} &= \left(1 - f_1^2(z_1^{[1]})\right) \frac{\partial \text{Div}}{\partial y_1^{[1]}} \\
\end{align*}
\]

Layer 1

\( K = 1 \)
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[ \frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688 \]
\[ \frac{\partial \text{Div}}{\partial z_1^{[2]}} = -0.1167 \]
\[ \frac{\partial \text{Div}}{\partial y_1^{[1]}} = -0.10509 \]
\[ \frac{\partial \text{Div}}{\partial y_2^{[1]}} = -0.03503 \]

\[ \frac{\partial \text{Div}}{\partial z_1^{[1]}} = f_1'(z_1^{[1]}) \frac{\partial \text{Div}}{\partial y_1^{[1]}} \]
\[ \frac{\partial \text{Div}}{\partial z_1^{[1]}} = \left(1 - f_1^2(z_1^{[1]})\right) \frac{\partial \text{Div}}{\partial y_1^{[1]}} \]
\[ \frac{\partial \text{Div}}{\partial z_1^{[1]}} = \left(1 - (y_1^{[1]})^2\right) \frac{\partial \text{Div}}{\partial y_1^{[1]}} \]
\[ \frac{\partial \text{Div}}{\partial z_1^{[2]}} = (1 - 0.1683^2)(-0.10509) \]

For \( k = N \), 1
For \( i = 1: \text{layer} - \text{width} \)
\[ \frac{\partial \text{Div}}{\partial z_i^{(k)}} = f_k'(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}} \]
\[ \frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]
\[ \frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]

Expected output: 1
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[
\begin{align*}
\frac{\partial \text{Div}}{\partial y_1^{[2]}} &= -0.4688 \\
\frac{\partial \text{Div}}{\partial z_2^{[2]}} &= (-0.1167) \\
\frac{\partial \text{Div}}{\partial y_1^{[1]}} &= -0.10509 \\
\frac{\partial \text{Div}}{\partial y_2^{[1]}} &= -0.03503
\end{align*}
\]

For \( k = N..1 \)

For \( i = 1: \text{layer} - \text{width} \)

\[
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f_k'(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = (-0.01966) \\
\frac{\partial \text{Div}}{\partial w_{21}^{[2]}} = 0.010481
\]
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \\
 z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \\
 z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[
\frac{\partial \text{Div}}{\partial y_2^{[1]}} = -0.4688 \\
\frac{\partial \text{Div}}{\partial z_2^{[1]}} = (-0.1167) \\
\frac{\partial \text{Div}}{\partial y_1^{[1]}} = -0.10509 \\
\frac{\partial \text{Div}}{\partial z_1^{[1]}} = -0.1021
\]

\[
\frac{\partial \text{Div}}{\partial z_1^{[1]}} = f_1'(z_1^{[1]}) \frac{\partial \text{Div}}{\partial y_1^{[1]}} \\
\frac{\partial \text{Div}}{\partial z_2^{[1]}} = \left(1 - f_1^2(z_2^{[1]})\right) \frac{\partial \text{Div}}{\partial y_2^{[1]}} \\
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = \left(1 - (y_2^{[1]})^2\right) \frac{\partial \text{Div}}{\partial y_2^{[1]}} \\
\frac{\partial \text{Div}}{\partial z_1^{[1]}} = (1 - 0.0897^2)(-0.03503)
\]

For \( k = N-1 \)
For \( i = 1: \text{layer} - \text{width} \)

\[
\frac{\partial \text{Div}}{\partial z_i^{[k]}} = f_k'(z_i^{[k]}) \frac{\partial \text{Div}}{\partial y_i^{[k]}} \\
\frac{\partial \text{Div}}{\partial y_i^{[k-1]}} = \sum_j w_{ij}^{[k]} \frac{\partial \text{Div}}{\partial z_j^{[k]}} \\
\frac{\partial \text{Div}}{\partial w_{ij}^{[k]}} = y_i^{[k-1]} \frac{\partial \text{Div}}{\partial z_j^{[k]}}
\]
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[
\begin{align*}
\frac{\partial \text{Div}}{\partial y_2^{[2]}} &= -0.4688 \\
\frac{\partial \text{Div}}{\partial y_1^{[1]}} &= -0.10509 \\
\frac{\partial \text{Div}}{\partial y_1^{[2]}} &= -0.03503 \\
\frac{\partial \text{Div}}{\partial z_1^{[1]}} &= -0.1021
\end{align*}
\]

For \( k = N-1 \)
For \( i = 1: \text{layer} - \text{width} \)
\[
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f_k'(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]
\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial z_2^{[1]}} = -0.0347
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = (-0.01966)
\]
\[
\frac{\partial \text{Div}}{\partial w_{21}^{[2]}} = 0.010481
\]
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[
\begin{align*}
\frac{\partial \text{Div}}{\partial y_1^{[2]}} &= -0.4688 \\
\frac{\partial \text{Div}}{\partial z_1^{[2]}} &= (-0.1167) \\
\frac{\partial \text{Div}}{\partial y_1^{[1]}} &= -0.10509 \\
\frac{\partial \text{Div}}{\partial z_1^{[1]}} &= -0.03503 \\
\frac{\partial \text{Div}}{\partial y_2^{[1]}} &= -0.1021 \\
\frac{\partial \text{Div}}{\partial z_2^{[1]}} &= -0.0347
\end{align*}
\]

\[ \partial \text{Div} \]

\[ \frac{\partial \text{Div}}{\partial w_{11}^{[1]}} = y_1^{[0]} \frac{\partial \text{Div}}{\partial z_1^{[1]}} \]
\[ \frac{\partial \text{Div}}{\partial w_{11}^{[1]}} = x_1 \frac{\partial \text{Div}}{\partial z_1^{[1]}} \]
\[ \frac{\partial \text{Div}}{\partial w_{11}^{[1]}} = 1.1(-0.1021) \]
\[ \frac{\partial \text{Div}}{\partial w_{11}^{[1]}} = -0.1123 \]
\[ \frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = (-0.01966) \]
\[ \frac{\partial \text{Div}}{\partial w_{21}^{[2]}} = 0.010481 \]

**For k = N-1**
**For i = 1:layer – width**

\[ \frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]

\[ \frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = f'_k(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}} \]

\[ \frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.4688 \quad \frac{\partial \text{Div}}{\partial z_2^{[2]}} = (-0.1167) \\
\frac{\partial \text{Div}}{\partial w_{12}} = -0.03503 \quad \frac{\partial \text{Div}}{\partial w_{22}} = -0.0347 \]

\[
\frac{\partial \text{Div}}{\partial w_{12}} = y_1^{[0]} \frac{\partial \text{Div}}{\partial z_2^{[1]}} \\
\frac{\partial \text{Div}}{\partial w_{22}} = x_1 \frac{\partial \text{Div}}{\partial z_2^{[1]}} \\
\frac{\partial \text{Div}}{\partial w_{12}} = 1.1(-0.0347) \\
\frac{\partial \text{Div}}{\partial w_{12}} = -0.03822 \]

\[
\frac{\partial \text{Div}}{\partial w_{11}} = (-0.01966) \quad \frac{\partial \text{Div}}{\partial w_{21}} = 0.010481 \quad \frac{\partial \text{Div}}{\partial w_{11}} = -0.1123 \]
Example: Backward

For $k = N..1$
For $i = 1:layer - width$

\[
\frac{\partial \text{Div}}{\partial z_i^{(k)}} = f'_k(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}}
\]

\[
\frac{\partial \text{Div}}{\partial w_{21}^{[1]}} = 0.4(-0.1021)
\]

\[
\frac{\partial \text{Div}}{\partial w_{21}^{[1]}} = -0.04084
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[1]}} = -0.01966
\]

\[
\frac{\partial \text{Div}}{\partial w_{12}^{[1]}} = 0.010481
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = -0.1123
\]

\[
\frac{\partial \text{Div}}{\partial w_{12}^{[2]}} = -0.03822
\]

\[
\frac{\partial \text{Div}}{\partial y_2^{[1]}} = -0.01021
\]

\[
\frac{\partial \text{Div}}{\partial y_1^{[1]}} = -0.10509
\]

\[
\frac{\partial \text{Div}}{\partial y_1^{[2]}} = -0.1167
\]

\[
\frac{\partial \text{Div}}{\partial y_2^{[2]}} = -0.03503
\]

\[
\frac{\partial \text{Div}}{\partial z_1^{[1]}} = -0.1021
\]

\[
\frac{\partial \text{Div}}{\partial z_2^{[1]}} = -0.0347
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = -0.01966
\]

\[
\frac{\partial \text{Div}}{\partial w_{12}^{[2]}} = 0.010481
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[1]}} = -0.1123
\]

\[
\frac{\partial \text{Div}}{\partial w_{12}^{[1]}} = -0.03822
\]

\[
\frac{\partial \text{Div}}{\partial w_{21}^{[1]}} = 0.4(-0.1021)
\]

\[
\frac{\partial \text{Div}}{\partial w_{21}^{[1]}} = -0.04084
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = -0.01966
\]

\[
\frac{\partial \text{Div}}{\partial w_{12}^{[2]}} = 0.010481
\]

\[
\frac{\partial \text{Div}}{\partial w_{11}^{[1]}} = -0.1123
\]

\[
\frac{\partial \text{Div}}{\partial w_{12}^{[1]}} = -0.03822
\]
Example: Backward

\[ z[1]_1 = 0.17 \quad y[1]_1 = 0.1683 \]
\[ z[1]_2 = -0.019 \quad y[1]_2 = -0.0897 \]
\[ z[2]_1 = 0.1246 \quad y[2]_1 = 0.5311 \]

\[
\frac{\partial \text{Div}}{\partial y[2]_1} = -0.4688 \quad \frac{\partial \text{Div}}{\partial z[2]_1} = (-0.1167)
\]
\[
\frac{\partial \text{Div}}{\partial w[21]_1} = y[0]_2 \frac{\partial \text{Div}}{\partial z[2]_1}
\]
\[
\frac{\partial \text{Div}}{\partial w[22]_1} = x_2 \frac{\partial \text{Div}}{\partial z[2]_1}
\]
\[
\frac{\partial \text{Div}}{\partial w[22]_1} = 0.4(-0.0347)
\]
\[
\frac{\partial \text{Div}}{\partial w[21]_1} = -0.10509 \quad \frac{\partial \text{Div}}{\partial w[21]_2} = -0.03503
\]
\[
\frac{\partial \text{Div}}{\partial w[22]_1} = -0.1021 \quad \frac{\partial \text{Div}}{\partial w[22]_2} = -0.0347
\]

For \( k = N..1 \)  
For \( i = 1: \text{layer} - \text{width} \)

\[
\frac{\partial \text{Div}}{\partial z[k]_i} = f'_k(z[k]_i) \frac{\partial \text{Div}}{\partial y[k]_i}
\]
\[
\frac{\partial \text{Div}}{\partial y[k-1]_i} = \sum_j w[k]_{ij} \frac{\partial \text{Div}}{\partial z[k]_j}
\]
\[
\frac{\partial \text{Div}}{\partial w[k]_{ij}} = y[k-1]_i \frac{\partial \text{Div}}{\partial z[k]_j}
\]

Expected output: 1

LAYER 1
K=1
Example: Backward

\[ z_1^{[1]} = 0.17 \quad y_1^{[1]} = 0.1683 \]
\[ z_2^{[1]} = -0.019 \quad y_2^{[1]} = -0.0897 \]
\[ z_1^{[2]} = 0.1246 \quad y_1^{[2]} = 0.5311 \]

\[ \frac{\partial \text{Div}}{\partial y_2^{[1]}} = -0.4688 \quad \frac{\partial \text{Div}}{\partial z_2^{[2]}} = (-0.1167) \]
\[ \frac{\partial \text{Div}}{\partial y_1^{[1]}} = -0.10509 \quad \frac{\partial \text{Div}}{\partial z_1^{[2]}} = -0.03503 \]
\[ \frac{\partial \text{Div}}{\partial z_1^{[1]}} = -0.1021 \quad \frac{\partial \text{Div}}{\partial z_2^{[1]}} = -0.0347 \]

\[ \frac{\partial \text{Div}}{\partial w_{11}^{[2]}} = (-0.01966) \quad \frac{\partial \text{Div}}{\partial w_{21}^{[2]}} = 0.010481 \]
\[ \frac{\partial \text{Div}}{\partial w_{11}^{[1]}} = -0.1123 \quad \frac{\partial \text{Div}}{\partial w_{12}^{[1]}} = -0.03822 \]
\[ \frac{\partial \text{Div}}{\partial w_{21}^{[1]}} = -0.04084 \quad \frac{\partial \text{Div}}{\partial w_{22}^{[1]}} = -0.013899 \]

For \( k = N \ldots 1 \)
For \( i = 1: \text{layer} - \text{width} \)

\[ \frac{\partial \text{Div}}{\partial z_i^{(k)}} = f'_k(z_i^{(k)}) \frac{\partial \text{Div}}{\partial y_i^{(k)}} \]
\[ \frac{\partial \text{Div}}{\partial y_i^{(k-1)}} = \sum_j w_{ij}^{(k)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]
\[ \frac{\partial \text{Div}}{\partial w_{ij}^{(k)}} = y_i^{(k-1)} \frac{\partial \text{Div}}{\partial z_j^{(k)}} \]
A real example
Digit classification

- MNIST dataset:
  - 70000 grayscale images of digits scanned.
  - 60000 for training
  - 10000 for testing

- Loss function

\[ J_2(w) = \frac{1}{m} \sum_{\text{train}} (\hat{y}_i - y_i)^2 \]
Digit classification

**Typical Problem statement:**

**multiclass classification**

- Given, many positive and negative examples (training data),
  - learn all weights such that the network does the desired job
A look in the code

• To run this code do:
  - import network
  - net = network.Network([784, 30, 10])
  - net.SGD(training_data, 30, 10, 3.0, test_data=test_data)
A look in code
A look in code
A look in the code

random Initialization
Feed forward ‘a’ thru all the layers

A Epoch is when all the training data has been used to update weights

A minibatch is a subset of all the data used to obtain a ‘quick’ weight updates

If there is test data perform evaluation
A look in the code

Add errors from all the training data from the mini-batch

Update the weights
references

• http://neuralnetworksanddeeplearning.com/chap1.html
• https://www.cs.cmu.edu/~bhiksha/courses/deeplearning/Fall.2015/
Questions?