Training Neural Networks

- Let us look at an example how to train a multilayer perceptron (MLP) neural network with sigmoid activations

- Define a regression loss

$$L(\theta) = \frac{1}{2M} \sum_{m=1}^{M} ||y^m - f_\theta(x^m)||^2$$

- Notation:

M: Number of data points

θ: Neural network parameters

y^m: d-dimensional label

x^m: Feature vector

Stochastic Gradient Descent

- To make the training more practical, stochastic gradient descent is used instead of standard gradient descent

- Recall that the idea of stochastic gradient descent is to approximate the gradient of a sum by sampling a few indices and averaging

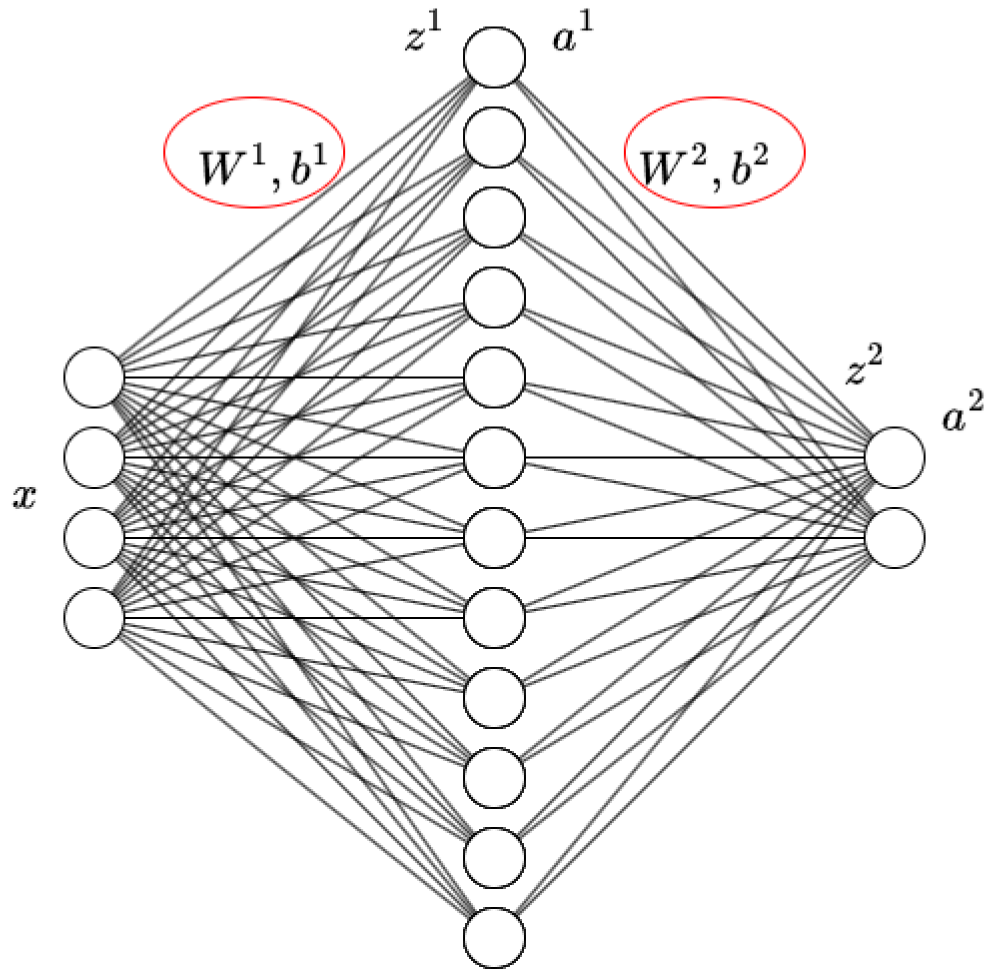
$$\nabla_\theta \sum_{i=1}^{n} L_i(\theta) \approx \frac{1}{K} \sum_{k=1}^{K} \nabla_\theta L_{i^k}(\theta)$$

- Here, each i^k is sampled uniformly at random from $\{1, \dots, n\}$

Learning the MLP Parameters

- We need to learn each W^l, b^l
  - Hence, we need their loss gradients!

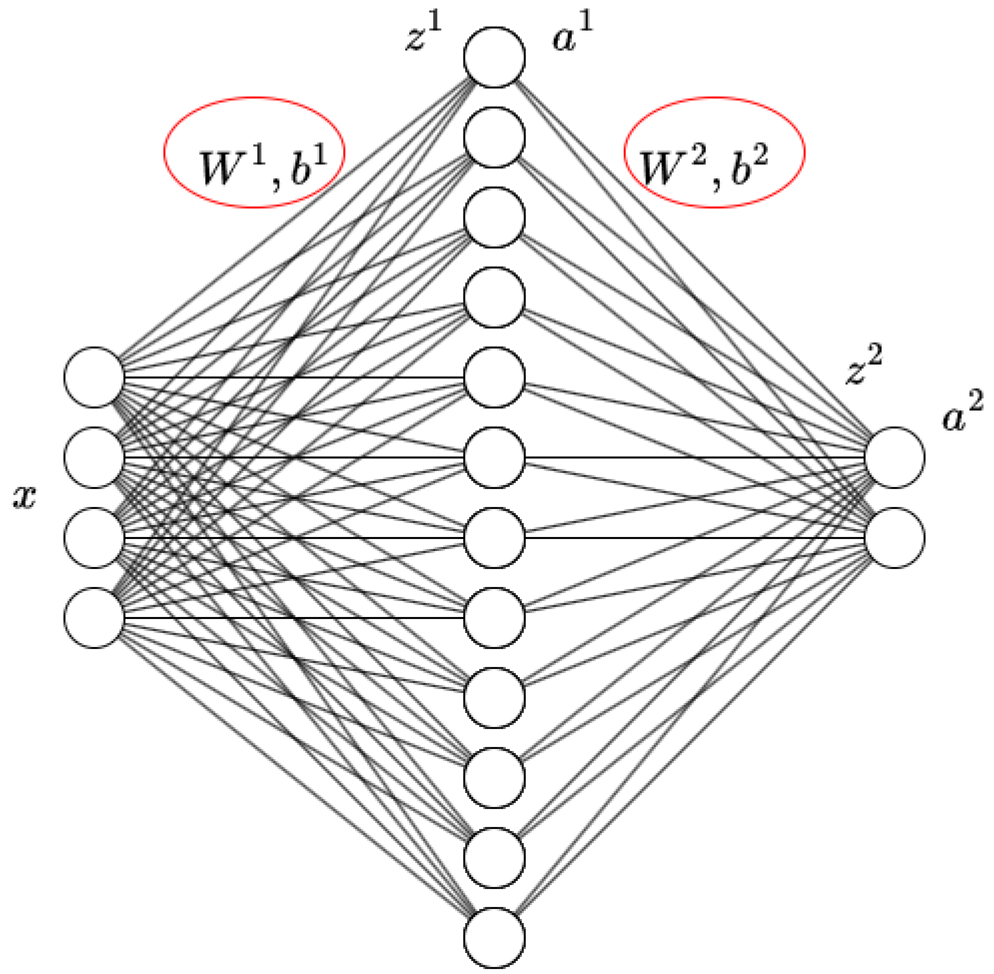
- To make it easier, let us define:
  - x: Input feature vector
  - W^l, b^l: Weight matrix, bias vector
  - z^l: Input to layer l's activation
  - a^l: Output of layer l's activation
  - L: Number of layers



Learning the MLP Parameters

- To make it easier, let us define:
  - x: Input feature vector
  - W^l, b^l: Weight matrix, bias vector
  - z^l: Input to layer l's activation
  - a^l: Output of layer l's activation
  - L: Number of layers

- We can express $a^L = \mathrm{f}_\theta(x)$ *recursively*
  - $a^2 = \sigma(z^2)$
  - $a^2 = \sigma(W^2 a^1 + b^2)$
  - $a^2 = \sigma(W^2 \sigma(z^1) + b^2)$
  - …



Learning the MLP Parameters

- We can also compute gradients recursively in terms of the input to each layer:
  - $\frac{\partial a^L}{\partial z^i} = \left[\frac{\partial a^L}{\partial z^L}\right]\left[\frac{\partial z^L}{\partial z^i}\right]$
  - $\frac{\partial a^L}{\partial z^i} = \left[\frac{\partial a^L}{\partial z^L}\right]\left[\frac{\partial z^L}{\partial a^{L-1}}\right]\left[\frac{\partial a^{L-1}}{\partial z^i}\right]$
  - $\frac{\partial a^L}{\partial z^i} = \left[\frac{\partial a^L}{\partial z^L}\right]\left[\frac{\partial z^L}{\partial a^{L-1}}\right]\left[\frac{\partial a^{L-1}}{\partial z^{L-1}}\right]\left[\frac{\partial z^{L-1}}{\partial z^i}\right]$
  - $\frac{\partial a^L}{\partial z^i} = \cdots$

$$\frac{\partial a}{\partial z} = \begin{bmatrix} \frac{\partial a_1}{\partial z_1} & \frac{\partial a_1}{\partial z_2} & \cdots & \frac{\partial a_1}{\partial z_m} \\ \frac{\partial a_2}{\partial z_1} & \frac{\partial a_2}{\partial z_2} & \cdots & \frac{\partial a_2}{\partial z_m} \\ \vdots & & & \\ \frac{\partial a_n}{\partial z_1} & \frac{\partial a_n}{\partial z_2} & \cdots & \frac{\partial a_n}{\partial z_m} \end{bmatrix}$$

A Jacobian matrix specifying partial derivatives

- Note: Each [] is a Jacobian matrix (multi-dimensional derivative)
  - MLP’s gradient computation is just repeated multivariate chain rule!
  - A sequence of Jacobian matrix products

Forming Backpropagation Strategy for MLPs

- Goal: Calculate $\frac{\partial L}{\partial W_{jk}^{l}}$ and $\frac{\partial L}{\partial b_{j}^{l}}$ for an input x

- What should our strategy be?
  - We note the following:

$$\frac{\partial L}{\partial W_{jk}^{l}} = \left[\frac{\partial L}{\partial z_j^l}\right]\left[\frac{\partial z_j^l}{\partial W_{jk}^{l}}\right] = \left[\frac{\partial L}{\partial z_j^l}\right]\left[\frac{\partial}{\partial W_{jk}^{l}}\left(W_{j*}^{l} a^{l-1} + b_j^l\right)\right] = \left[\frac{\partial L}{\partial z_j^l}\right] a_k^{l-1}$$

$$\frac{\partial L}{\partial b_{j}^{l}} = \left[\frac{\partial L}{\partial z_j^l}\right]\left[\frac{\partial z_j^l}{\partial b_{j}^{l}}\right] = \left[\frac{\partial L}{\partial z_j^l}\right]\left[\frac{\partial}{\partial b_{j}^{l}}\left(W_{j*}^{l} a^{l-1} + b_j^l\right)\right] = \left[\frac{\partial L}{\partial z_j^l}\right]$$

Forming Backpropagation Strategy for MLPs

- Goal: Calculate $\frac{\partial L}{\partial W_{jk}^l}$ and $\frac{\partial L}{\partial b_j^l}$ for an input x

- What should our strategy be?
  - We note the following:

$$\frac{\partial L}{\partial W_{jk}^l} = \left[\frac{\partial L}{\partial z_j^l}\right]\left[\frac{\partial z_j^l}{\partial W_{jk}^l}\right] = \left[\frac{\partial L}{\partial z_j^l}\right]\left[\frac{\partial}{\partial W_{jk}^l}\left(W_{j*}^l a^{l-1} + b_j^l\right)\right] = \left[\frac{\partial L}{\partial z_j^l}\right] a_k^{l-1}$$

$$\frac{\partial L}{\partial b_j^l} = \left[\frac{\partial L}{\partial z_j^l}\right]\left[\frac{\partial z_j^l}{\partial b_j^l}\right] = \left[\frac{\partial L}{\partial z_j^l}\right]\left[\frac{\partial}{\partial b_j^l}\left(W_{j*}^l a^{l-1} + b_j^l\right)\right] = \left[\frac{\partial L}{\partial z_j^l}\right]$$

  - We can get our gradients this way if we repeatedly solve the highlighted for each layer!

Forming Backpropagation Strategy for MLPs

- We can solve for $\frac{\partial L}{\partial z^l}$ recursively!
  - $\frac{\partial L}{\partial z^l} = \left[\frac{\partial L}{\partial z^L}\right]\left[\frac{\partial z^L}{\partial z^l}\right] = \left[\frac{\partial L}{\partial a^L}\right]\left[\frac{\partial a^L}{\partial z^L}\right]\left[\frac{\partial z^L}{\partial z^l}\right] = \delta^L \left[\frac{\partial z^L}{\partial z^l}\right]$
  - $\frac{\partial L}{\partial z^l} = \delta^L \left[\frac{\partial z^L}{\partial a^{L-1}}\right]\left[\frac{\partial a^{L-1}}{\partial z^{L-1}}\right]\left[\frac{\partial z^{L-1}}{\partial z^l}\right] = \delta^{L-1} \left[\frac{\partial z^{L-1}}{\partial z^l}\right]$
  - $\frac{\partial L}{\partial z^l} = \delta^{L-1} \left[\frac{\partial z^{L-1}}{\partial a^{L-2}}\right]\left[\frac{\partial a^{L-2}}{\partial z^{L-2}}\right]\left[\frac{\partial z^{L-2}}{\partial z^l}\right] = \delta^{L-2} \left[\frac{\partial z^{L-2}}{\partial z^l}\right]$
  - …

- Each $\delta^l = \frac{\partial L}{\partial z^l}$, which is what we need!

- Q: What is the recursive definition for δ^l?

Forming Backpropagation Strategy for MLPs

- We can solve for $\frac{\partial L}{\partial z^l}$ recursively!
  - $\frac{\partial L}{\partial z^l} = \left[\frac{\partial L}{\partial z^L}\right]\left[\frac{\partial z^L}{\partial z^l}\right] = \left[\frac{\partial L}{\partial a^L}\right]\left[\frac{\partial a^L}{\partial z^L}\right]\left[\frac{\partial z^L}{\partial z^l}\right] = \delta^L \left[\frac{\partial z^L}{\partial z^l}\right]$
  - $\frac{\partial L}{\partial z^l} = \delta^L \left[\frac{\partial z^L}{\partial a^{L-1}}\right]\left[\frac{\partial a^{L-1}}{\partial z^{L-1}}\right]\left[\frac{\partial z^{L-1}}{\partial z^l}\right] = \delta^{L-1} \left[\frac{\partial z^{L-1}}{\partial z^l}\right]$
  - $\frac{\partial L}{\partial z^l} = \delta^{L-1} \left[\frac{\partial z^{L-1}}{\partial a^{L-2}}\right]\left[\frac{\partial a^{L-2}}{\partial z^{L-2}}\right]\left[\frac{\partial z^{L-2}}{\partial z^l}\right] = \delta^{L-2} \left[\frac{\partial z^{L-2}}{\partial z^l}\right]$
  - …

- The above tells us that
  - $\delta^L = \left[\frac{\partial L}{\partial a^L}\right]\left[\frac{\partial a^L}{\partial z^L}\right]$
  - $\delta^{l-1} = \delta^l \left[\frac{\partial z^l}{\partial a^{l-1}}\right]\left[\frac{\partial a^{l-1}}{\partial z^{l-1}}\right]$

← All that's left is to solve for these derivatives!

Base Step: Computing δ^L

- $\delta^L = \left[\frac{\partial L}{\partial a^L}\right]\left[\frac{\partial a^L}{\partial z^L}\right]$
- $\delta^L = \left[\frac{\partial}{\partial a^L}\left(||y - a^L||^2\right)\right]\left[\frac{\partial a^L}{\partial z^L}\right]$
- $\delta^L = [-2(y - a^L)]^T\left[\frac{\partial a^L}{\partial z^L}\right]$
- $\delta^L = [-2(y - a^L)]^T\left[\frac{\partial}{\partial z^L}\sigma(z^L)\right]$
- $\delta^L = [-2(y - a^L)]^T\left[\mathrm{diag}\left(\sigma(z^L) \circ \left(1 - \sigma(z^L)\right)\right)\right]$

$$\frac{\partial \sigma(x)}{\partial x} = \sigma(x)\left(1 - \sigma(x)\right)$$

We can derive a vectorized version by applying sigmoid element-wise

The resulting Jacobian is a diagonal matrix with the above derivative on the diagonal for each vector element

Recursive Step: Computing δ^l

- $\delta^{l-1} = \delta^l \left[\frac{\partial z^l}{\partial a^{l-1}}\right]\left[\frac{\partial a^{l-1}}{\partial z^{l-1}}\right]$
- $\delta^{l-1} = \delta^l \left[\frac{\partial}{\partial a^{l-1}}(W^l a^{l-1} + b^l)\right]\left[\frac{\partial a^{l-1}}{\partial z^{l-1}}\right]$
- $\delta^{l-1} = \delta^l [W^l] \left[\frac{\partial a^{l-1}}{\partial z^{l-1}}\right]$
- $\delta^{l-1} = \delta^l [W^l] \left[\frac{\partial}{\partial z^{l-1}}\left(\sigma\left(z^{l-1}\right)\right)\right]$
- $\delta^{l-1} = \delta^l [W^l] \left[\mathrm{diag}\left(\sigma\left(z^{l-1}\right) \circ \left(1 - \sigma\left(z^{l-1}\right)\right)\right)\right]$

Full Backpropagation for MLPs

1. Compute the inputs/outputs for each layer by starting at the input layer and applying sigmoids (forward pass)

2. Compute δ^L for the output layer

$$\delta^L = [-2(y - a^L)]^T \left[\mathrm{diag}\left(\sigma(z^L) \circ \left(1 - \sigma(z^L)\right)\right)\right]$$

3. Starting from $l = L - 1$ and working backwards, compute (backward pass)

$$\delta^{l-1} = \delta^l [W^l] \left[\mathrm{diag}\left(\sigma\left(z^{l-1}\right) \circ \left(1 - \sigma\left(z^{l-1}\right)\right)\right)\right]$$

4. Perform gradient descent

$$b_j^l = b_j^l - \gamma \cdot \delta_j^l$$

$$W_{jk}^l = w_{jk}^l - \gamma \cdot \delta_j^l a_k^{l-1}$$