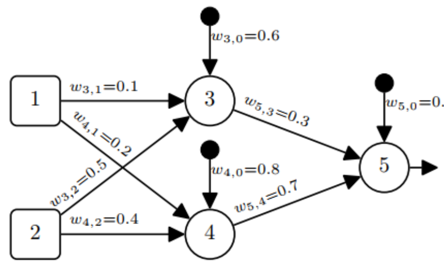
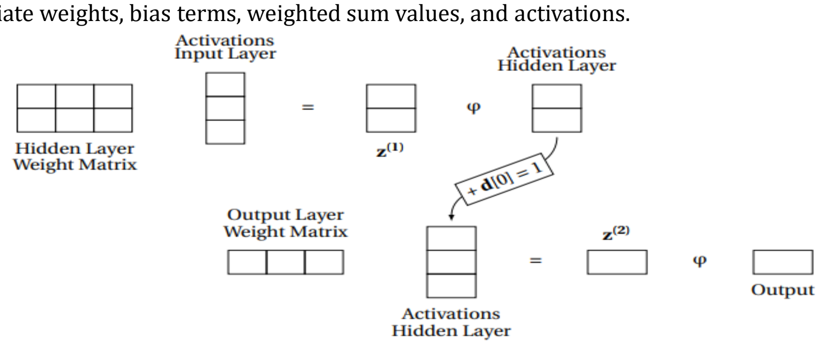
Machine Learning
Tutorial 6 (Deep Learning)

Exercise1:

The following image shows an artificial neural network with two sensing neurons (Neurons 1 and 2) and 3 processing neurons (Neurons 3, 4, and 5).

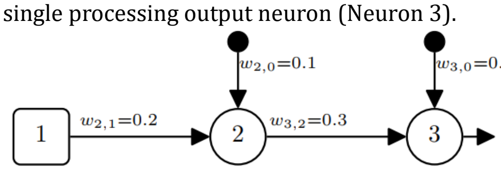


(a) Assuming that the processing neurons in this network use a **logistic** activation function, what would be the output of Neuron 5 if the network received the input vector: Neuron 1 = 0.7 and Neuron 2 = 0.3?
(b) Assuming that the processing neurons in this network use a **ReLU** activation function, what would be the output of Neuron 5 if the network received the input vector: Neuron 1 = 0.7 and Neuron 2 = 0.3?
(c) The following image provides a template diagram for the sequence of matrix operations that our neural network would use to process the input vector Neuron 1 =0.7 and Neuron 2 = 0.3. Assuming that the processing neurons in the network use a ReLU activation function, fill in the diagram with the appropriate weights, bias terms, weighted sum values, and activations.



Exercise 2:

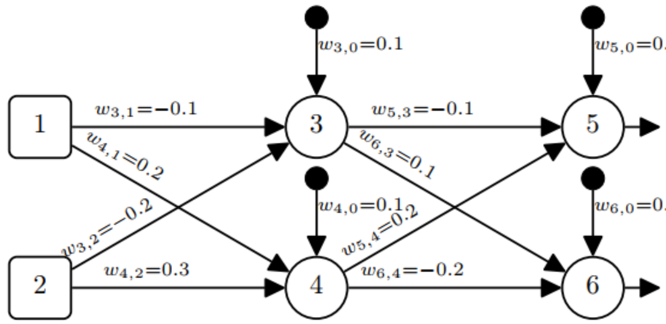
The following image illustrates the topology of a simple feedforward neural network that has a single sensing neuron (Neuron 1), a single hidden processing neuron (Neuron 2), and a single processing output neuron (Neuron 3).



(a) Assuming that the processing neurons use **logistic** activation functions, that the input to the network is Neuron 1 = 0.2 and that the desired output for this input is 0.7:
i. Calculate the output generated by the network in response to this input.
ii. Calculate the δ values for each of the neurons in the network (i.e., δ3, δ2).
iii. Using the δ values you calculated above, calculate the sensitivity of the error of the network to changes in each of the weights of the network (i.e., $\partial\mathcal{E}/\partial w_{3,2}$, $\partial\mathcal{E}/\partial w_{3,0}$, $\partial\mathcal{E}/\partial w_{2,1}$, $\partial\mathcal{E}/\partial w_{2,0}$).
iv. Assuming a learning rate of α = 0.1, calculate the updated values for each of the weights in the network (*w*3,2; *w*3,0; *w*2,1; *w*2,0) after the processing of this single training example.
v. Calculate the reduction in the error of the network for this example using the new weights, compared with using the original weights.
(b) Assuming that the processing neurons are **ReLU**s, that the input to the network is Neuron 1 = 0.2, and that the desired output for this input is 0.7:
i. Calculate the output generated by the network in response to this input.
ii. Calculate the δ values for each of the neurons in the network (i.e., δ3, δ2).
iii. Using the δ values you have calculated in the preceding, calculate the sensitivity of the error of the network to changes in each of the weights of the network (i.e., $\partial\mathcal{E}/\partial w_{3,2}$, $\partial\mathcal{E}/\partial w_{3,0}$, $\partial\mathcal{E}/\partial w_{2,1}$, $\partial\mathcal{E}/\partial w_{2,0}$).
iv. Assuming a learning rate of α = 0.1, calculate the updated values for each of the weights in the network (*w*3,2; *w*3,0; *w*2,1; *w*2,0) after the processing of this single training example.
v. Calculate the reduction in the error for this example using the new weights for the network, compared with using the original weights.

Exercise 3:

The following image illustrates the topology of a feedforward neural network that has two sensing neurons (Neurons 1 and 2), two hidden processing neuron (Neurons 3, and 4), and two processing output neurons (Neurons 5 and 6).



Assuming that the processing neurons use **logistic** activation functions, that the input to the network is Neuron 1 = 0.3 and Neuron 2 = 0.6, and that the desired output for this input is Neuron 5 = 0.7 and Neuron 6 = 0.4:

i. Calculate the output generated by the network in response to this input.

ii. Calculate the sum of squared errors for this network for this example.

iii. Calculate the δ values for each of the processing neurons in the network (i.e., δ_6, δ_5, δ_4, δ_3).

iv. Using the δ values you calculated above, calculate the sensitivity of the error of the network to changes in each of the weights of the network (i.e., (i.e., $\partial\mathcal{E}/\partial w_{6,4}$, $\partial\mathcal{E}/\partial w_{6,3}$, $\partial\mathcal{E}/\partial w_{6,0}$, $\partial\mathcal{E}/\partial w_{5,4}$, $\partial\mathcal{E}/\partial w_{5,3}$, $\partial\mathcal{E}/\partial w_{5,0}$, $\partial\mathcal{E}/\partial w_{4,2}$, $\partial\mathcal{E}/\partial w_{4,1}$, $\partial\mathcal{E}/\partial w_{4,0}$, $\partial\mathcal{E}/\partial w_{3,2}$, $\partial\mathcal{E}/\partial w_{3,1}$, $\partial\mathcal{E}/\partial w_{3,0}$).

v. Assuming a learning rate of α = 0.1, calculate the updated values for each of the weights in the network (w6,4; w6,3; w6,0; w5,4; w5,3; w5,0; w4,2; w4,1; w4,0; w3,2; w3,1; w3,0) after the processing of this single training example.

vi. Calculate the reduction in the sum of squared error of the network for this example using the new weights, compared with using the original weights.

Exercise 4:

Assuming a fully connected feedforward network where all the neurons use a linear activation function (i.e. $a_i = z_i$) and with the following topology:

(a) 100 neurons in the input layer
(b) 5 hidden layers with 2,000 neurons in each layer
(c) 10 neurons in the output layer

If all the inputs to the network have been standardized to have a mean value of 0 and a standard deviation of 1, and the initial weights for the network are sampled from a normal distribution with mean 0.0 and standard deviation of σ = 0.01, then:

(a) Calculate the variance of the *z* values across for the neurons in the first hidden layer in the first iteration of training.
(b) Calculate the variance of the *z* values across for the neurons in the last hidden layer in the first iteration of training.

Assuming that the variance of the δs for the output layer is equal to 1:

(a) Calculate the variance of the δs across for the neurons in the last hidden layer in the first iteration of training.
(b) Calculate the variance of the δs values across for the neurons in the first hidden layer in the first iteration of training.
(c) Is the training dynamic of this network stable, or is it suffering from vanishing or exploding gradients?

Exercise 5:

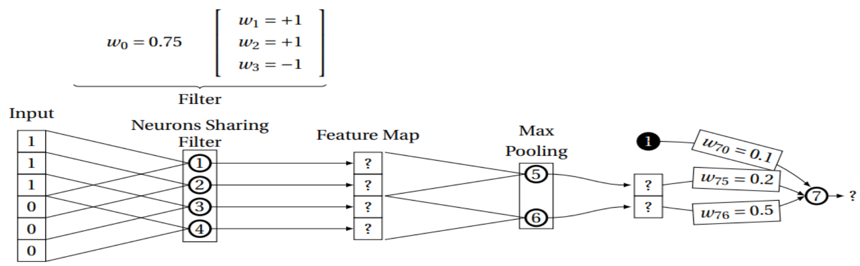
Assuming a network architecture that has four neurons in a **softmax** output layer. If the one-hot encoding of the target for the current training example is **t** = [0, 0, 1, 0] and the logits for the four neurons in the softmax output layer for this example are [0, 0.5, 0.25, 0.75], then what is the δ value for each of the four neurons?

Exercise 6:

Assuming a feedforward neural network that has 4 neurons in hidden layer *k* and that we are training this network using **inverted dropout** with ρ = 0.5. If the activations for the neurons in layer *k* are as follows: [0.2, 0, 4, 0, 3, 0.1] and the **DropMask** for layer *k* is [1, 0, 1, 0], calculate the activation vector that is actually propagated to layer *k* = 1 after inverted dropout has been applied.

Exercise 7:

The figure below illustrates a layer of a convolutional neural network that is processing a one-dimensional input. For ease of reference each of the neurons in the network has been labeled: 1, 2, 3, 4, 5, 6, 7. The architecture of the network consists of ReLUs that share a filter (Neurons 1, 2, 3, 4), followed by a sub-sampling layer containing two max pooling units (Neurons 5, 6), and then a fully connected layer containing a single ReLU (Neuron 7). The ReLU in the first layer has a 3-by-1 receptive field, and there is a stride of 1 used in this layer. The max pooling units have a receptive field of 2-by-1, and there is no overlap between the receptive fields of the max pooling units.



(a) What value will this network output?

(b) Assuming the target output for this input is 1, calculate the δ for each neuron in the network.

(c) Calculate the weight update for each weight in the filter: $w0$, $w1$, $w2$, $w3$.

Exercise 8:

Assume a simple recurrent neural network architecture matching the one shown in the detailed schematic on the left of Figure 8.37. This network has two input neurons, three ReLUs in the hidden layer, and two ReLUs in the output layer. Also, all the bias terms in the network are equal to 0.1, and the weight matrices of the network (excluding bias terms) are listed in Equation (8.5).

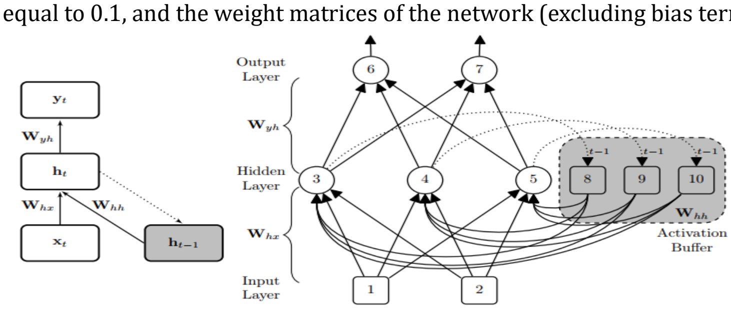


Figure 8.37
Schematic of the simple recurrent neural architecture.

$$\underbrace{\begin{bmatrix} -0.07 & 0.05 \\ -0.04 & 0.1 \\ -0.05 & -0.05 \end{bmatrix}}_{\mathbf{W}_{hx}} \underbrace{\begin{bmatrix} -0.22 & -0.1 & 0.05 \\ -0.04 & -0.09 & -0.06 \\ -0.09 & 0 & -0.08 \end{bmatrix}}_{\mathbf{W}_{hh}} \underbrace{\begin{bmatrix} 0.06 & -0.01 & -0.18 \\ -0.06 & 0.13 & 0.14 \end{bmatrix}}_{\mathbf{W}_{yh}} \quad (8.5)$$

(a) If $\mathbf{x}_t = [1, 0.5]$ and $\mathbf{h}_{t-1} = [0.05, 0.2, 0.15]$, calculate the value of $\mathbf{y}_t$.

(b) Assuming that the target output for time t_t = [0, 1], calculate the δ value for each neuron in the network.