ELEC 5970/6970 Special Topics: AI and Neuromorphic Hardware

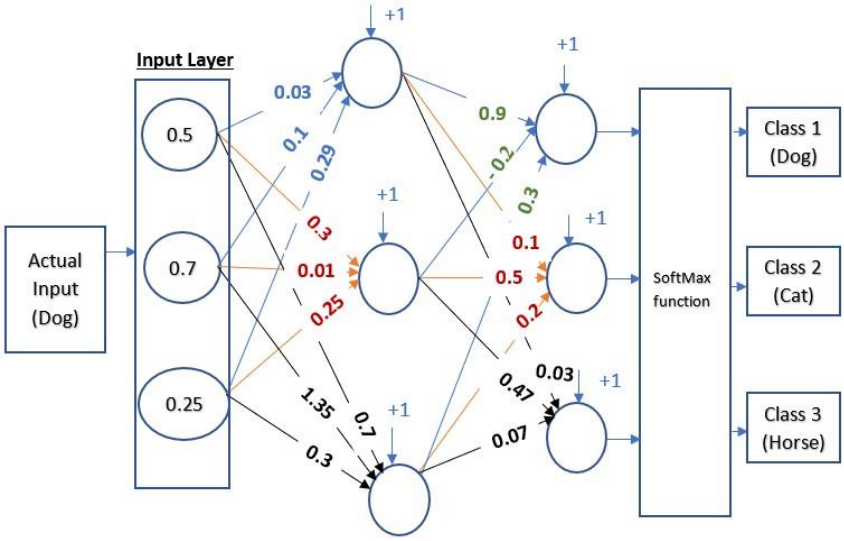
Spring 2022

Homework/ Lab1

Problem 1:

(a) Calculate the **cross-entropy loss** of the following network when the input image is an image of **Dog**. Show all your calculations. Use Matrix formats. Use RELU activation.

(b) What is activation function. What are the benefits of using RELU over Sigmoid?



1. Class 1 = y_1, class 2 = y_2, class 3 = y_3

$a'_1 = x_1 w'_{11} + x_2 w'_{12} + x_3 w'_{13}$

$a'_2 = x_1 w'_{21} + x_2 w'_{22} + x_3 w'_{23}$

$a'_3 = x_1 w'_{31} + x_2 w'_{32} + x_3 w'_{33}$

$y_1 = a'_1 w^2_{11} + a'_2 w^2_{12} + a'_3 w^2_{13}$

$y_2 = a'_1 w^2_{21} + a'_2 w_{22} + a'_3 w_{23}$

$y_3 = a'_1 w^2_{31} + a'_2 w_{32} + a'_3 w_{33}$

$a'_1 = (0.5)(0.03) + (0.7)(0.1) + (0.25)(0.24) + 1 = 1.1575$

$a'_2 = (0.5)(0.3) + (0.7)(0.01) + (0.25)(0.25) + 1 = 1.2195$

$a'_3 = (0.5)(0.7) + (0.7)(1.35) + (0.25)(0.8) + 1 = 2.35$

$y_1 = (1.1575)(0.9) + (1.2195)(0.2) + (2.37)(0.3) + 1 = 2.5088$

$y_2 = (1.1575)(0.1) + (1.2195)(0.5) + (2.37)(0.2) + 1 = 2.19$

$y_3 = (1.1575)(0.03) + (1.2195)(0.47) + (2.37)(0.07) + 1 = 1.$

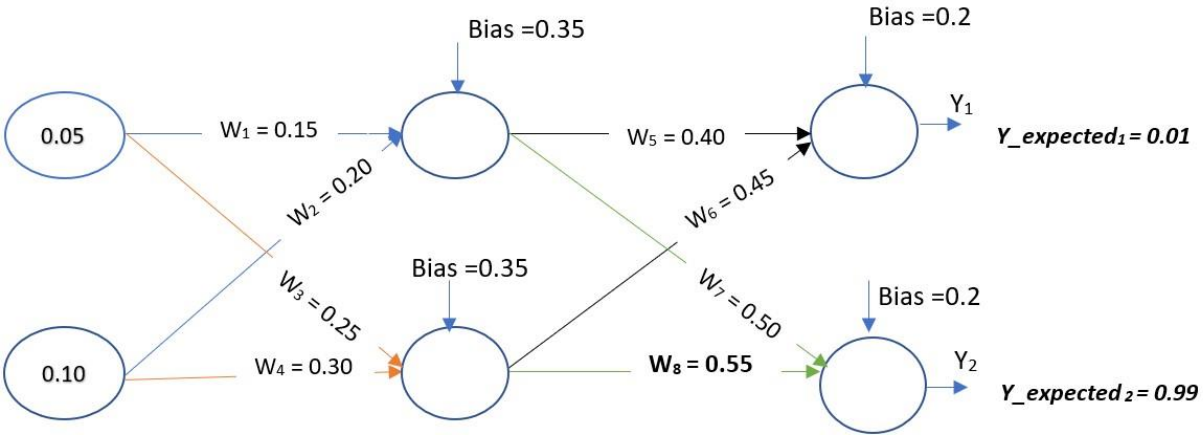
passing y_1, y_2, y_3 to Softmax

$e^{2.5088} = 12.290787\,52$ $\quad e^{2.1995} = 9.0205$ $\quad e^{1.773} = 5.89$

$y_1 = \frac{12.29}{27.20} = 0.4518$ $\quad y_2 = \frac{9.02}{27.2} = 0.33$

$y_3 = \frac{5.89}{27.2} = 0.22$

Problem 2: Find updated $\mathbf{W_8^*}$ after one backpropagation. Use **ReLU** activation function and **MSE cost** function to find the total error. Learning rate =0.01. (Note training set size =1, no need to use soft-max at output]. Show your calculations.


Bias =0.35
Bias =0.2
0.05
W1 = 0.15
W5 = 0.40
Y1
Y_expected1 = 0.01
W2 = 0.20
W6 = 0.45
Bias =0.35
W3 = 0.25
W7 = 0.50
Bias =0.2
0.10
W4 = 0.30
W8 = 0.55
Y2
Y_expected2 = 0.99

2. $w_{8\,new} = w_{8\,old} - \gamma \frac{\delta E}{\delta w_8} \;/\; E = \frac{1}{2}\left[(y^1_{real} - y^1_{pred})^2 + (y^2_{real} - y^2_{pred})^2\right]$

$a^1_1 = (0.05)(0.15) + (0.10)(0.20) + 0.35 = 0.3775$

$a'_1 = Relu(0.3775) = 0.3775$

$a^1_2 = (0.05)(0.25) + (0.10)(0.30) + 0.35 = 0.3925$

$a'_2 = Relu(0.3925) = 0.3925$

$a^2_1 = (0.3775)(0.40) + (0.3925)(0.45) + 0.2 = 0.527625$

$a^2_2 = (0.3775)(0.5) + (0.3925)(0.55) + 0.2 = 0.604625$

$E = \frac{1}{2}\left[(0.01 - 0.528)^2 + (0.99 - 0.605)\right]$

$E = 0.327$

$\delta z^3_2 = \frac{\delta}{\delta w_8}\left[(a'_1 \times w_7) + (a'_2 \times w_8) + 0.2 = a'_2\right]$

$\frac{\delta y_2}{\delta z^3_2} = 1$ partial w/ respect to itself $= 1$

$\frac{\delta E}{\delta y_2} = y^2_{real} - y^2_{pred} = 0.99 - 0.605 = 0.39$

$\frac{\delta E}{\delta w_8} = 0.39 \times 1 \times 0.39$

$\frac{\delta E}{\delta w_8} = 0.15$

$w_{8\,new} = 0.55 - 0.01(0.151)$

$w_{8\,new} = 0.5485$

Problem 3:

What is training epoch? To obtain an accuracy of 97.89% in 20 epochs, the weights of a model were updated 10,000 times. If the mini-batch size during training was 512, find the number of training samples the model was trained on.

(a)

- Epoch is equivalent to the number of passes of the complete training dataset while training the model.
- For gradient Decent, using an appropriate loss function, error for all the samples is calculated and one epoch has only single weight update.
- For stochastic gradient descent, one epoch has weight updates equals the size of the training dataset.
- For batch/mini-batch gradient decent, the training samples are divided into mini-batches and the number of weight updates in one epoch equals training dataset size/ mini-batch size

(b)

- In 20 epochs, weight updates 10,000 times
  So, in 1 epoch weight updates = (10,000/20) = 500 times.

training samples = number of times weights are updated x size

- 500 * 512 = 256,000

Problem 4:

Quantize the following weight matrix into **int8** datatype. Write the corresponding matrix showing only the int8 converted values in decimal. Assume zero_point =0.

$$\begin{bmatrix} 2.9 & 0.59 & -1.35 \\ 1.5 & 1.47 & -0.75 \\ 0.7 & 0.35 & 2.5 \end{bmatrix}$$

Problem 5

What is overfitting? State and briefly describe the techniques you can use to avoid overfitting.

Overfitting is when the model fits exactly against the training data (IBM). Therefore the algorithm is unprotected against unseen data which ultimately defeats its purpose. Overfitting happens when the model begins to learn the irrelevant information in the dataset so it cannot perform classification or predictions. We can use Cross Validation or Dropout to reduce overfitting.

Problem 6

What is Pruning in context of DNN? What is the motivation behind Pruning (discuss from hardware point of view)? What is the difference between Dropout and Pruning?

Pruning reduces the size of neural networks and therefore requires less storage, operates at a higher speed, and has less power dissipation. Pruning gets rid of weights with very low magnitudes and occurs but then needs to be retrained to regain some of the lost accuracy. We can use Dropout to mitigate overfitting and occurs during training.

Problem 7

What is the motivation behind BFloat16 data type? What are the advantages of BFloat16 data type over FP32 data type?

Bfloat16 is Google's data type which is 16 bits and covers the same as FP32 but it uses half the number of bits. It is more compact and does more with less bits. The ability to do more with less bits can save data and time.

Problem 8

(a) What is the difference between Inference and Training? (b) What is int8 quantization? (c) Can you use int8 quantization during Training? If not, explain why. (d) Explain how Quantization-aware-training can improve inference accuracy.

- Inference is when the model is already trained and functioning. With Training, the model starts with no knowledge.
- Int8 quantization is a tool used to increase the resiliency of neural networks to noise. For training you should not use int8, only for inference.
- Int8 during training will cause significant accuracy loss because the errors caused by int8 will increase.
- Quantization-aware-training can improve inference by being used in weight and bias matrices.

Problem 9:

What are the problems of using a fixed learning rate? Explain the motivations of using ADAM optimization method.

If the learning rate is set too low, the training will progress very slowly as you are making miniscule updates to the weights in your network. However, if the learning rate is set too high, it can cause unwanted divergent behavior in the loss function.