CIE6032 and MDS6232: Homework #1

Due on Sunday, 2021/03/07, 23:59 pm

Problem 1 [15 points]

Cross entropy is often used as the objective function when training neural networks in classification problems. Suppose the training set includes N training pairs $D=\left\{\left(x_i^{(\text{train})}, y_i^{(\text{train})}\right)\right\}_{i=1}^{N}$, where $x_i^{(\text{train})}$ is a training sample and $y_i^{(\text{train})} \in \{1,\ldots,c\}$ is its corresponding class label. $\mathbf{z}_i$ is the output of the network given input $x_i^{(\text{train})}$ and the nonlinearity of the output layer is softmax. $\mathbf{z}_i$ is a c dimensional vector, $z_{i,k} \in [0,1]$ and $\sum_{k=1}^{c} z_{i,k} = 1$. The questions are as follows.

(1) Write the objective function of cross-entropy with softmax activation function, and calculate the gradient of hidden-to-output and input-to-hidden weights as we introduced in class. **[10 points]**

(2) Verify it is equivalent to the negative log-likelihood on the training set, assuming the training samples are independent. **[5 points]**

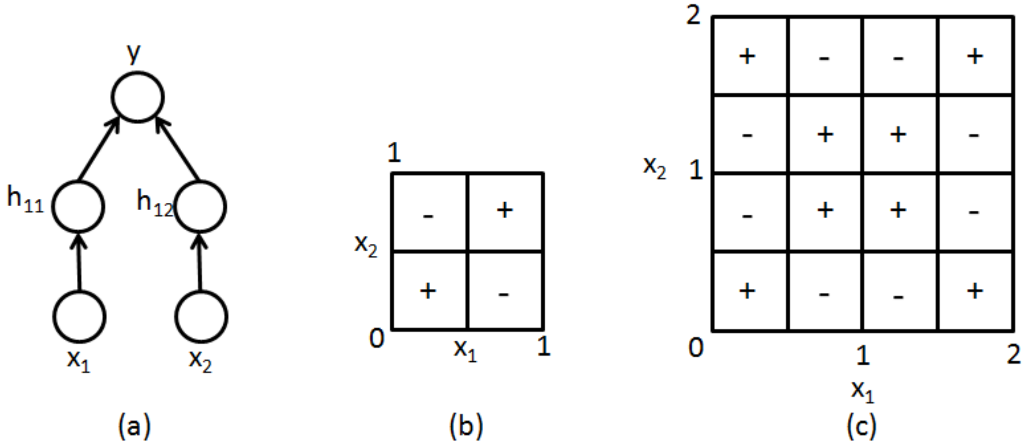
Problem 2 [25 points]

x_1 and x_2 are two input variables, and y is the target variable to be predicted. The network structure is shown in Figure 1(a). $h_{11} = f_{11}(x_1)$, $h_{12} = f_{12}(x_2)$, and $y = g(h_{11}, h_{12})$. The questions are as follows.

(1) Assuming $x_1 \in [0,1]$ and $x_2 \in [0,1]$, in order to obtain the decision regions in Figure 1(b), decide functions f_{11}, f_{12}, and g. **[5 points]**

(2) Now we extend the range of x_1 and x_2 to [0, 2]. Please add one more layer to Figure 1(a) in order to obtain the decision regions in Figure 1(c). **[5 points]**

(3) Although the decision boundaries in Figure 1(c) look complicated, there exist regularity and global structure. Please explain such regularity and global structure. Based on your observation, draw the decision boundaries when the range of x_1 and x_2 to [0, 4]. **[5 points]**

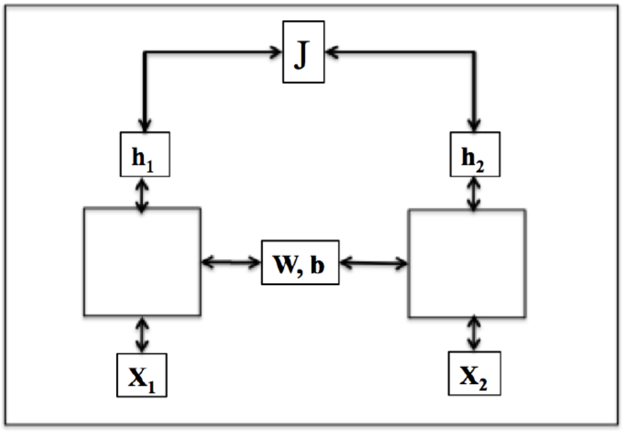


(4) Following the question above and assuming the range of x_1 and x_2 is extended to $\left[0,2^n\right]$, draw the network structure and the transform function in each layer, in order to obtain the decision regions with the same regularity and global structure in Figure 1 (b) and (c). The complexity of computation units should be $O(n)$. **[5 points]**

(5) Assuming the range of x_1 and x_2 is $\left[0,2^n\right]$ and only one hidden layer is allowed, specify the network structure and transform functions. **[5 points]**

Problem 3 [30 points]

Siamese neural nets have an interesting architecture– the same parameters and functions are used to evaluate 2 inputs. As one might expect, Siamese nets are useful to train **similarity** metrics, evaluations of how “close” inputs are. These nets have been applied to facial recognition tasks with a good deal of success, but in this example, we’ll see how to train Siamese nets to learn a distance metric for two inputs, e.g., word vectors. One might imagine training a net to map word vectors across languages, discover synonyms or antonyms, etc.



Here is one such model to evaluate how similar two inputs are using Euclidean distance. There are two inputs $x_1, x_2 \in R^n$, shared parameters $W \in R^{m\times n}$ and $b \in R^m$, and a single hidden layer associated with each input:

$$h_1 = \sigma\left(Wx_1 + b\right)$$

$$h_2 = \sigma\left(Wx_2 + b\right)$$

We evaluate the distance between the two activations h_1, h_2 using Euclidean distance as our similarity metric. The model objective J is

$$J = \frac{1}{2}\left\|h_1 - h_2\right\|_F^2 + \frac{\lambda}{2}\left\|W\right\|_F^2$$

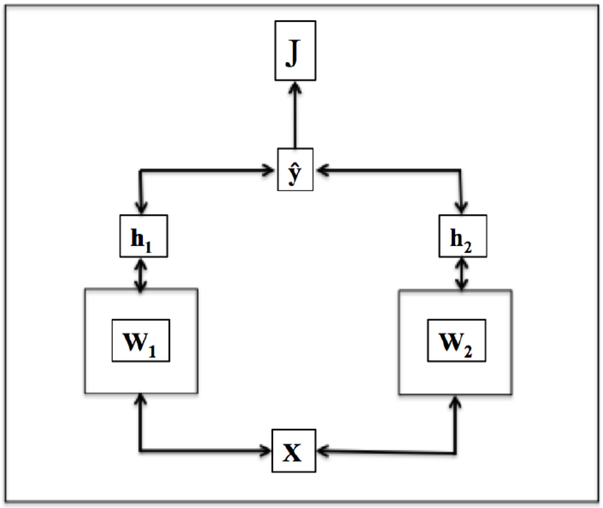
where λ is a given regularization parameter. (The Frobenius norm $\left\|\bullet\right\|_F$ is a matrix norm defined by $\left\|\mathbf{A}\right\|_F = \sqrt{\sum_{i,j}\left|A_{ij}\right|^2}$). The questions are as follows.

(1) Calculate the gradient $\nabla_W J$ and $\nabla_b J$. **[5 points]**

(2) Write out the (vanilla) gradient descent update rules for the model parameters for a single training example (with arbitrary step size α). **[5 points]**

(3) If $W \in R^{10\times 5}$ and $b \in R^{10\times 1}$, how many parameters does the model have? **[5 points]**

Now imagine you wanted to see how ReLU/sigmoid nonlinearities might affect training on **single** input. But instead of training two separate nets, you want to train a psuedo-Siamese net like the one below.



The whole model would be changed to

$$h_1 = \sigma\left(Wx + b_1\right)$$

$$h_2 = \mathrm{Relu}\left(Wx + b_2\right)$$

$$\hat{y} = soft\max(W_3(h_1 + h_2) + b_3),$$

where $x \in R^n$, $W_1, W_2 \in R^{m\times n}$, $W_3 \in R^{k\times m}, b_1, b_2 \in R^m$ and $b_3 \in R^k$. We calculate this model for N examples and k classes with cross-entropy loss

$$J = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{k} y_j^i \log(\hat{y}_j^i),$$

where y_j is the one-hot vector for example j with all probability mass on the correct class and $\hat{y}_j$ are the softmax scores for example j.

(4) Calculate $\nabla_{h_1} J, \nabla_{h_2} J,$ and $\nabla_x J$. **[10 points]**

(5) For W_1, W_2, which one is likely to train faster, please explain it. **[5 points]**

Problem 4 [30 points]

(1) Finish the Coding/Planar_MLP.ipynb according to requirements, i.e., fill in code in required lines **[15 ponits]**

(2) According to the tutorial introduced on course, please rewrite the MLP model defined from scratch in Planar_MLP.ipynb using Pytorch/Mxnet and powerful autograd function. **[15 ponits]**