E9 205 – Machine Learning for Signal Processing

Homework # 3

March 25, 2025

Due date: April 9, 2025

Analytical part, prepared in writing, can be scanned. This should be attached to the report on the coding part. Finally, a single pdf file containing the response is to be submitted. Source code also needs to be included.
Name of file should be "Assignment3_FullName.pdf" submitted to teams channel.
Assignment should be solved individually without collaboration with other human or online (GPT-like) resources..

1. Download the larger IMDB sentiment analysis dataset from `http://leap.ee.iisc.ac.in/sriram/teaching/MLSP25/assignments/data/IMDB_Dataset.csv.zip`.

   Split the data randomly into train (40000), test (5000) and remaining for validation.

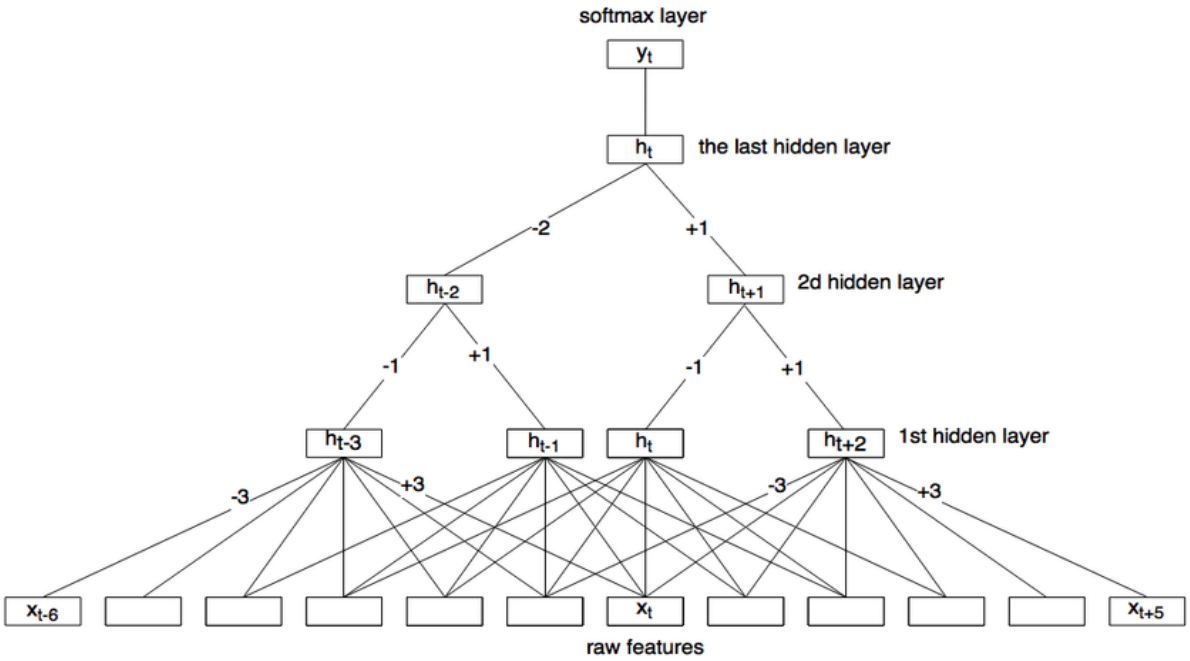
   (a) Use the pre-trained word2vec model (*gensim* is a popular tool for loading pre-trained word2vec models) to generate 100 dimensional word embeddings for each sentence. Train a PCA model for converting 100 dimensional embeddings to 2 dimensions (using the training data) and make a scatter plot of the word embeddings for words from 10 validation sentences. Report any semantic clustering of words that you observe in this plot.

   (b) Using the word2vec embeddings as features, train and evaluate an LSTM model with the following configuration, 2 hidden layers, with 256 cells, followed by average pooling and one-classification layer, trained with Binary Cross Entropy (BCE) to classify the sentiment classes. Train the model for 25 epochs with a batch-size 32 using the Adam optimizer. Plot the train, and validation loss curves, and measure the accuracy on the test data.

   (c) Replace the average pooling layer in the above question with an attention based pooling after the 2-layer LSTM model. Train the model for 25 epochs with a batch-size 32 using the Adam optimizer. Plot the train, validation loss curves, and measure the accuracy on the test data. Compare and contrast the model's performance and behavior with model explored in (b).

   (d) Replace the two-layer LSTM model with a single transformer encoder layer with 256 hidden dimensions, followed by average pooling and classification with BCE loss. Plot the train, validation loss curves, and measure the accuracy on the test data. Contrast the model's performance and behavior with models explored in (b) and (c).

   (**Points** 40)



2. **Time Delay NN** While working on music information retrieval task with data $\mathbf{x}_t$, Sam is proposing a new architecture, based on time delay networks as shown above,

   The numbers indicated along the weights are context of the inputs used in the subsequent layer. Only the solid arrows shown in the figure indicate connections. The output units have a softmax activation function and hidden units realize a ReLU activation function without any bias variable.

   For this TDNN architecture, can you help Sam compute the number of learnable parameters and the back propagation update for each of these parameters. The loss is,

   $$\mathcal{E} = \sum_t C.E(\mathbf{y}_t, \mathbf{z}_t) \tag{1}$$

   where $C.E$ is cross entropy and $\{\mathbf{z}_t\}_{t=1}^{T}$ is the one hot target sequence. (**Points 20**)

3. Download the data listed in `http://leap.ee.iisc.ac.in/sriram/teaching/MLSP25/assignments/data/ESC-50-master.zip`

   Use the only the data corresponding to ESC-10 part (400 files) (defined in meta/esc50.csv file). Use Folds-1,2,3 for training (241 files for training), fold-4 for validation (80 files) and fold-5 for testing (79 files). Extract mel-spectrogram features from the audio with 128 frequency channels (dimensions) for all the 5 second files and 10ms window shift. Thus, each file would be a matrix of 128×500.

   Make a CNN model architecture for this 10-class classification setting with the following details, two layers of 2-D CNN with 16 filters 3×3 size, with stride of 1×1 and with max-pooling of 3×3. Flatten the CNN outputs and use 2 fully connected layers of hidden dimensions 128 and then classification with softmax non-linearity for 10 classes. Use the cross-entropy loss for training the models. Use a batch-size of 16 for all these experiments.

(a) Compare the training and validation loss curves for training with (a) SGD, (b) SGD with momentum (factor of 0.9), (c) RMSprop (with default parameters) and (d) Adam optimizer. Report the test accuracy for each of these configurations as well.

(b) At the flattened output and at the input of 2 fully connected layers, compare the training and validation loss curves for a) No-norm, b) Layer norm and c) Batch norm. Use the Adam optimizer for all these experiments.

(c) Train three different CNN models on the training data. For the first model, use SGD training without any normalization. For the second model use the RMSprop optimizer with Layer norm. For the third model, use Adam optimizer with Batch norm. Ensemble the model outputs using a) output averaging of the posterior model outputs from the three model outputs, and b) with optimal linear weighted combination of the three model outputs (use the validation data to derive the optimal model combination weights). Comment on the performance gains obtained using model ensembling.

(**Points** 30)

4. Note that a CNN layer realizes a convolutional, optional pooling and non-linearity layer.

   (i) Comment on the computational complexity and memory requirements of a general CNN layer with that of a feedforward DNN layer.

   (ii) Show how a CNN layer can be realized as a feedforward layer with weight sharing and sparse connections. Conversely, construct a CNN model layer, that acts like a dense feedforward neural layer.

   (**Points 10**)