Quiz-05

- Due Sep 29 at 11:59pm
- Points 10
- Questions 10
- Available Sep 27 at 6pm - Sep 29 at 11:59pm
- Time Limit None
- Allowed Attempts 3

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	151 minutes	7 out of 10

⚠ Correct answers are hidden.

Score for this attempt: 7 out of 10
Submitted Sep 28 at 4:59pm
This attempt took 151 minutes.

Question 1
1 / 1 pts

Consider the Following Hypothetical:

You've always been a proponent of the Standard Normal Distribution for weight initialization. However, after delving into Kaiming He's paper, you're taken aback by the efficacy of his scaling approach compared to the traditional normal distribution initialization. Eager to validate these findings, you meticulously recreate the 22-layer model as described in the paper, employing He's initialization. In a twist of fate, **all the weights of the network are divided by sqrt(2) post-initialization**. Based on the insights from the paper, **which outcome is the most plausible following this modification**?

Refer to the detailed analysis in the paper: https://arxiv.org/pdf/1502.01852.pdf (https://arxiv.org/pdf/1502.01852.pdf)

- The network converges slightly slower than what was claimed in the paper but has poorer accuracy.
- The network converges slightly slower than what was claimed in the paper but the final accuracy is not very different.

The sqrt(2) division converts the Kaiming initialization to a Xavier initialization. As per the paper, for a 22-layer variant Xavier converges slower than Kaiming but the two are not very different in

terms of their final accuracies.

- The network gradients explode to large values leading to NaN.
- The network stalls and makes very little progress due to diminishing gradients.

Question 2
1 / 1 pts

In a neural network, a specific subset of the parameters $(w_1, w_2, w_3, \cdots, w_L)$ are all constrained to be identical in value, i.e. $w_1 = w_2 = w_3 \cdots = w_L = w^S$. Which of the following represents a valid SGD pseudocode to update their values when the network is being trained. Here T is the number of training instances in the batch, X_t is the t-th training instance and Loss(X_t) is loss computed for the t-th training instance. For each instance assume the forward and backward passes (to compute derivatives) have been performed before the "for i = 1:L" loop. Also assume $dLoss(X_t)/dw_i$ is computed using backdrop.

Hint: **(Lecture 9: Slides 65 - 74)**

for t = 1:T

$w^S = 0$

for i = 1:L

$w_i = w_i - \eta * dLoss(X_t)/dw_i$

$w^S = w^S + w_i$

for i = 1:L

- $w_i = w^S$

for t = 1:T

for i = 1: L

- $w_i = w_i - \eta * dLoss(X_t)/dw_i$

for t = 1:T

for i = 1:L

$w^S = w^S - \eta * dLoss(X_t)/dw_i$

for i = 1:L

- $w_i = w^S$

for t = 1:T

for i = 1:L

$w^S = w^S - \eta * dLoss(X_t)/dw_i$

$w_i = w^S$

Question 3
1 / 1 pts

You are given the problem of “wake-up word detection” -- the problem of recognizing if the wake-up word “IDLgod” has been spoken in a recording. You know “IDLgod” takes less than half a second to say, so you build a little “IDLgod” classification MLP that can analyze a half-second segment of speech and decide if the wake-up word has been spoken in it. You plan to scan incoming audio in chunks of half a second, sliding forward 25 milliseconds at a time, to detect if the wake-up word has occurred in any chunk.
Unfortunately, the training data you are given only consists of longish 15-20 second long segments of audio. The recordings tagged as “positive” data have the word “IDLgod” spoken somewhere in them, but the precise location is not marked. You are not even informed *how many* times the word is spoken in the recording -- all you know is that somewhere in that long recording are one or more instances of the wake-up word. So you have no way of slicing out the precise half-second segments where the word was spoken, to train your MLP, should you decide to do so.

The negative recordings do not have the word in them at all.

How would you train a model using this data? Keep in mind that during test time, you want to be able to classify incoming audio in chunks of exactly half a second.

Hint: **(Lecture 9: Slides 4 - 46)**

Slice up the input into half-second segments, and treat all segments derived from the positive recordings as positive instances to train your MLP.

Manually inspect the positive instances, and find the precise boundaries of the wake-up word (and extract 0.5sec segments of audio that contain them) to “clean up” your training data and train your model.

For each input of T seconds, construct a large network with (about) 40T copies of the MLP, with all of their outputs going into a softmax. Analyze the entire input with the large network, such that that the i-th copy of the MLP within the large network “looks” at the half second section of time starting at t = ((i-1)25+1) ms. Train the entire network over the complete recordings using KL divergence w.r.t. the labels for the inputs, using gradient descent, where all the copies of the MLP within the larger network are constrained to share parameters (i.e. have identical parameters)

For each input of T seconds, construct a large network with (about) 40T copies of the MLP, with all of their outputs going into a softmax. Analyze the entire input with the large network, such that that the i-th copy of the MLP within the large network “looks” at the half second section of time starting at t = ((i-1)25+1) ms. Train the entire network over the complete recordings using KL divergence w.r.t. the labels for the inputs using regular gradient descent, where gradients of network parameters are computed using backprop.

IncorrectQuestion 4
0 / 1 pts

Which of these pseudocodes perform the operations of a TDNN?

Hint:

Recall that by that we mean it scans a 1D instance comprising a time series with T vectors for patterns. Don’t worry about edge cases or shapes : assume that all these code are syntactically correct and represent different networks, and that all variables Y are preallocated to size sufficiently greater than T to ensure no exceptions happen in the code, and that these preallocated values of Y (other than the input to the net) are initialized to 0.

In the codes below, the index “l” represents the layer index. The networks have L layers. N represents the number of filters in a layer and K is the width of the filters.

Assume Y(0,*) is the input to the network, where “*” represents the necessary set of variables required to represent the entire input. The number of indices represented must be inferred from the code. In the notation below, in the products of w and Y, when both are multi-dimensional arrays (tensors), the notation represents a scalar valued tensor-inner-product, where corresponding components of the w and Y tensors are multiplied, and the set of pair-wise products is added to result in a scalar.

```
for t = 1:T
  for l = 1:L
    Y(l,t) = activation(w(t)Y(l-1,t:t+K))
```

```
Y = softmax({Y(L,:,:)})
```

```
for l = 1:L
  for t = 1:T
    for m = 1:N
      Y(l,m,t) = activation(w(l,m,:,:,)Y(l-1,:,t:t+K))
Y = softmax({Y(L,:,:)})
```

```
for l = 1:L
  for t = 1:T
    Y(l,t) = activation(w(l)Y(l,t-1))
Y = softmax({Y(L,:,:)})
```

```
for t = 1:T
  for l = 1:L
    for m = 1:N
      Y(l,m,t) = activation(w(l,m,:,:,)Y(l-1,:,t:t+K))
Y = softmax({Y(L,:,:)})
```

Question 5
1 / 1 pts
Mark all statements that are true.

Hint: Lecture 10, slides 16-38

- [x] The convolutional layers in Lecun's CNN model the processing of S-layers discovered by Hubel and Wiesel
- [x] Yann Lecun's CNN is a supervised analog of Fukushima's Neocognitron
- [x] The sequence of processing steps in Lecun's CNN is derived biological inspiration from Hubel and Wiesel's studies
- [x] The pooling layers in Lecun's CNN model the processing of the C-layers discovered by Hubel and Wiesel

IncorrectQuestion 6
0 / 1 pts

A time-delay neural network has three convolutional layers followed by a softmax unit. The convolutional layers have the following structure:

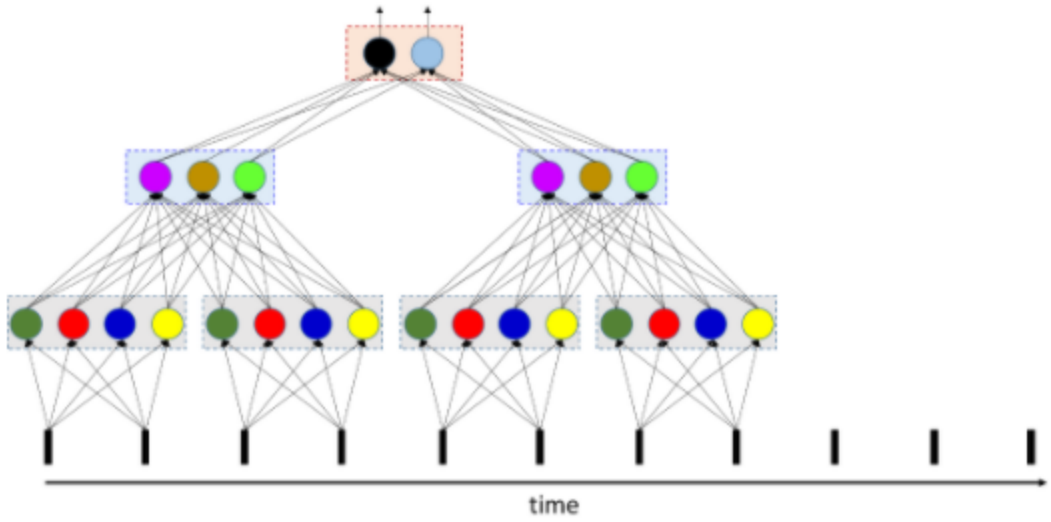
- The first hidden layer has 4 filters of kernel-width 2;
- The second layer has 3 filters of kernel-width 2;
- The third layer has 2 filters of kernel-width 2.

Assume that the stride of the convolution is 2 in every layer. As explained in class, the convolution layers of this TDNN are exactly equivalent to scanning the input with a (shared-parameter) MLP (and passing the set of outputs of the MLP at the individual time instants through a final softmax). What would be the architecture of this scanning MLP?

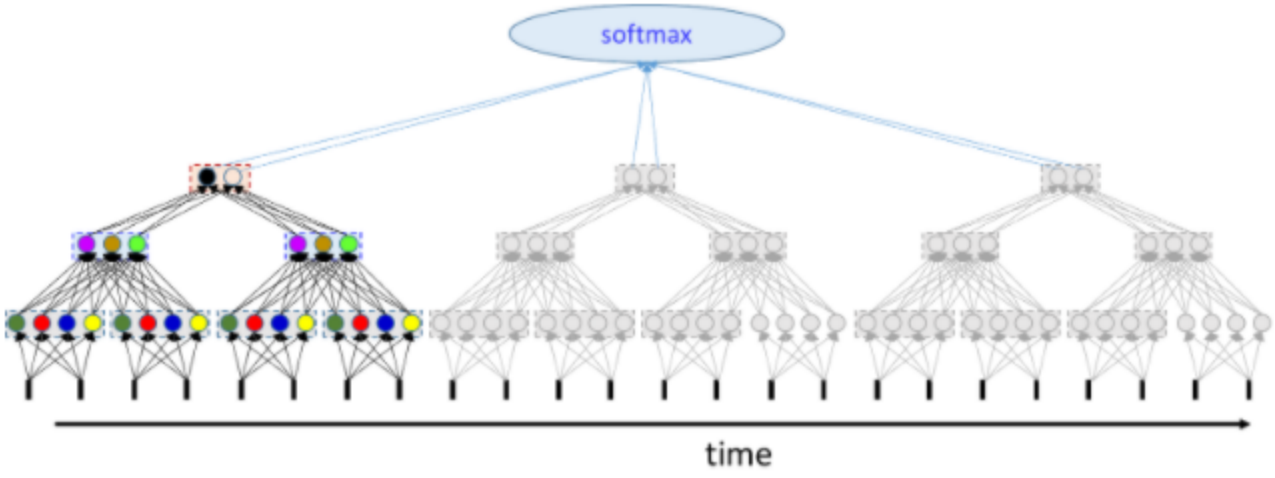
Hint: Piazza @659

- [ ] A three layer MLP with 4 neurons in the first layer, 3 in the second layer and 2 in the third layer
- [ ] Classification with this TDNN cannot be modeled as scanning with an MLP
- [x] A three layer MLP with 8 neurons in the first layer, 6 neurons in the second layer and 4 neurons in the third layer
- [ ] A three-layer MLP with 16 neurons in the first layer, 6 in the second layer and 2 in the third layer
- [ ] A one-layer MLP with a layer of size 24

The basic MLP is shown below. The sequence of little black bars shows the time sequence of input vectors. All blocks with the same background color are identical and share parameters (weights and biases). Neurons represented using the same color are identical and share parameters.



The full scan would have the structure shown below:



Question 7
1 / 1 pts

A time-delay neural network has three convolutional layers followed by a softmax unit. The convolutional layers have the following structure:

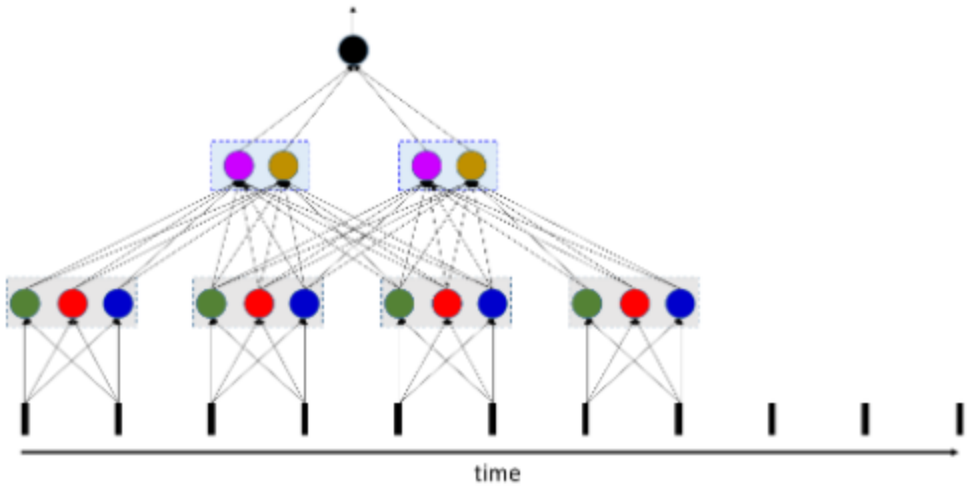
- The first hidden layer has 3 filters of kernel-width 2 and convolutions with a stride of 2;
- The second layer has 2 filters of kernel-width 3 and convolutions with a stride of 1;
- The third layer has 1 filter of kernel-width 2 and convolutions with a stride of 1.

As explained in class, the convolution layers of this TDNN are exactly equivalent to scanning the input with a shared-parameter MLP (and passing the set of outputs of the MLP at the individual time instants through a final softmax). What would be the architecture of this MLP?

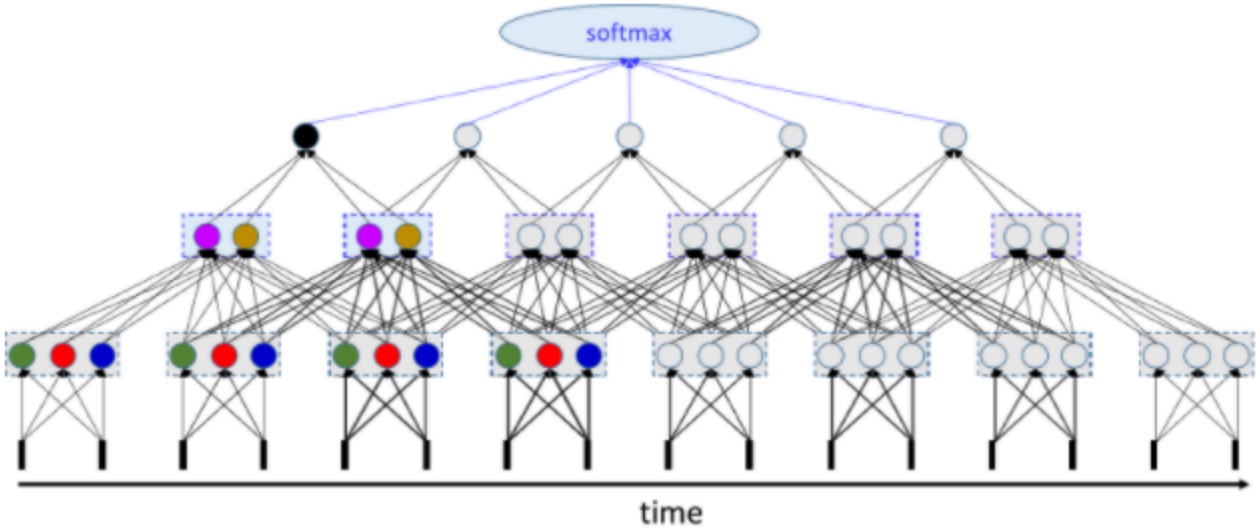
Hint: Piazza @659

- [ ] A three-layer MLP with 3 neurons in the first layer, 2 in the second, and 1 in the third
- [x] A three-layer MLP with 12 neurons in the first layer, 4 in the second, and 1 in the third
- [ ] A three-layer MLP with 6 neurons in the first layer, 6 in the second, and 2 in the third
- [ ] Classification with this TDNN cannot be modeled as scanning with an MLP
- [ ] A three-layer MLP with 12 neurons in the first layer, 4 in the second, and 2 in the third

The basic MLP is shown below. The sequence of little black bars shows the time sequence of input vectors. All blocks with the same background color are identical and share parameters (weights and biases). Neurons represented using the same color are identical and share parameters.



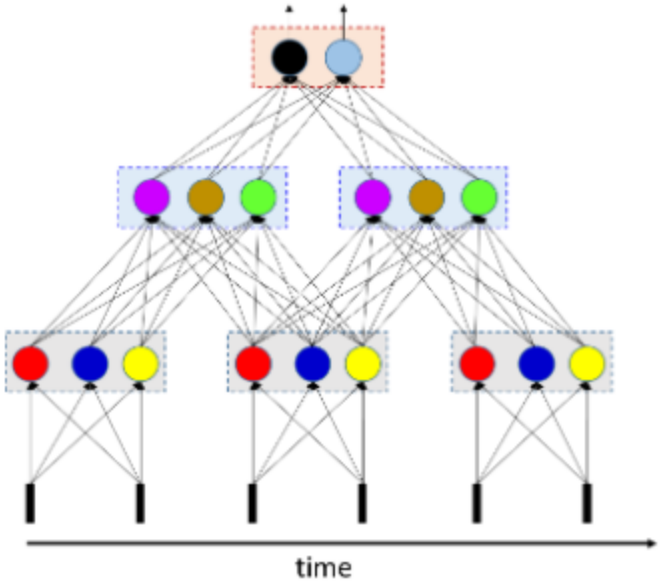
The full scan would have the structure shown below:



IncorrectQuestion 8

0 / 1 pts

A time-series input is scanned for a pattern using the following MLP. While scanning, the entire network strides two time steps at a time. Subsequently the outputs of the MLP at each time step are combined through a softmax. In the figure, neurons of identical color within each layer have identical responses.



The same operation can be performed using a time-delay neural network. What will the architecture of this TDNN be?

Hint: Piazza @659

- [ ] Three convolutional layers. 9 filters of kernel-width 2 in the first layer, with stride 2, 6 filters of kernel-width 2 in the second layer, with stride 2, and 2 filters of kernel-width 2 in the final layer with stride 1.
- [ ] Two convolutional layers and a flat layer. 9 filters of kernel-width 2 in the first layer, with stride 2. 6 filters of kernel-width 2 with stride 3 in the second layer. The final flat layer has two neurons with all the outputs of the second layer going to them.
- [ ] Three convolutional layers. 3 filters of kernel-width 2 in the first layer, with stride 2, 3 filters of kernel-width 2 in the second layer, with stride 1, and 2 filters of kernel-width 2 in the third layer, with a stride of 1.
- [x] Three convolutional layers. 3 filters of kernel-width 2 in the first layer, with stride of 2, 3 filters of kernel-width 6 with stride 3 in the second layer, and 2 filters of kernel-width 6 with stride 3 in layer 3.

Question 9
1 / 1 pts

Which of the following best describe the "shift-invariance" that a Convolutional Neural Network may have?

Hint: Lecture 9, Slide 9-21

- (•) The model's response to an image and a translated version of that image will be the same
- ( ) The model's response to an image and a rotation of that image will be the same
- ( ) The model's response to an image and a translated version of that image will be the same, as well as the model's response to an image and a magnified version of that image will be the same
- ( ) The model's response to an image and a magnified version of that image will be the same

Question 10
1 / 1 pts

Consider an RGB image input ***I***, and a filter ***K*** of size 7x7. On convolving image ***I*** with filter ***K,*** which of the following statements is true?

Hint: How does the number of output channels relate to the number of filters and the number of input channels?

- ( ) The output is 3-dimensional but the depth is not 7
- ( ) The output is 3-dimensional with depth 7
- (•) The output is 2-dimensional
- ( ) The output is 7-dimensional

Quiz Score: 7 out of 10