GOURAB BHATTACHARYYA - 17004888

CSE-512-HW3

1>

① Given that, $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

$$H(x) = \text{sgn}(f(x))$$

$$\epsilon_{training} = \frac{1}{N} \sum_{j=1}^{N} \delta(H(x^j) \neq y^j)$$

Therefore, $\epsilon_{training}$ = # of incorrect predictions

we know that,

if $f(x^j) > 0$, then predict 1 and predict -1 otherwise

So, $f(x^j) y^j > 0$, for correct predictions

and $f(x^j) y^j < 0$, for incorrect predictions

Now, since $e^{-f(x^j) y^j} < 1$, for correct predictions

and $e^{-f(x^j) y^j} > 1$, for incorrect predictions

Let, $\delta_1 \geq 1$

$\delta_2 \in (0, 1)$

Therefore,

$$N\epsilon_{training}\,\delta_1 \geq N\epsilon_{training}$$

as $\delta_2 \in (0, 1)$, so we can do the following,

$$(N - N\epsilon_{training})\,\delta_2 + N\epsilon_{training}\,\delta_1 \geq N\epsilon_{training} \quad \text{—①}$$

From the previous explanation given, we can re-write,

$$(N - N\epsilon_{training})\,\delta_2 + N\epsilon_{training}\,\delta_2 = \sum_{j=1}^{N} e^{-f(x^j)y^j}$$

From equation —①

$$N\epsilon_{training} \leq \sum_{j=1}^{N} e^{-f(x^j)y^j}$$

$$\Rightarrow \boxed{\epsilon_{training} \leq \frac{1}{N} \sum_{j=1}^{N} e^{-f(x^j)y^j}}$$

(Hence proved)

1)

(2) From previous problem its given that,

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

Starting from left hand side;

$$\frac{1}{N} \sum_{j=1}^{N} e^{-f(x^j) y^j}$$

$$= \frac{1}{N} \sum_{j=1}^{N} e^{-\sum_{t=1}^{T} \alpha_t h_t(x^j) y^j}$$

$$= \frac{1}{N} \sum_{j=1}^{N} \prod_{t=1}^{T} e^{-\alpha_t h_t(x^j) y^j}$$

$$= \frac{1}{N} \sum_{j=1}^{N} \prod_{t=1}^{T} \frac{w_j^{(t+1)} Z_t^*}{w_j^t}$$ [by using given equation, $w_j^{(t+1)} = \frac{w_j^t \exp(-\alpha_t y^j h_t(x^j))}{Z_t}$]

$$= \frac{1}{N} \sum_{j=1}^{N} \frac{w_j^{(t+1)}}{w_j^t}$$

$$= \frac{1}{N} \sum_{j=1}^{N} \frac{w_j^{(T+1)}}{w_j^1} \prod_{t=1}^{T} Z_t$$

$$= \frac{1}{N} \sum_{j=1}^{N} \frac{w_j^{(T+1)}}{1/N} \prod_{t=1}^{T} Z_t$$ [as we know, $w_j^1 = 1/N$ as an initial condition for normalized weights]

(P.T.O.)

$$\frac{1}{N}\sum_{j=1}^{N} e^{-f(x^j)y^j} = \sum_{j=1}^{N} w_j^{(T+1)} \prod_{t=1}^{T} Z_t$$

$$= 1 \cdot \prod_{t=1}^{T} Z_t \quad \left[\text{since weights are normalized, so, } \sum_{j=1}^{N} w_j^{(T+1)} = 1\right]$$

$$\Rightarrow \boxed{\frac{1}{N}\sum_{j=1}^{N} e^{-f(x^j)y^j} = \prod_{t=1}^{T} Z_t}$$

$$\Rightarrow \boxed{\text{L.H.S} = \text{R.H.S}}$$

(Hence proved)

1)

3.a)

Given,

$$Z_t = (1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t}$$

Taking derivative w.r.t α_t, we get,

$$\frac{\partial Z_t}{\partial \alpha_t} = -(1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t}$$

by setting this equation to zero (0),

$$-(1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t} = 0$$

$$\Rightarrow (1-\epsilon_t)e^{-\alpha_t} = \epsilon_t e^{\alpha_t}$$

$$\Rightarrow e^{2\alpha_t} = \frac{1-\epsilon_t}{\epsilon_t}$$

$$\Rightarrow \boxed{\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)}$$ (Answer)

Therefore,

$$Z_t^{opt} = (1-\epsilon_t)\sqrt{\frac{\epsilon_t}{(1-\epsilon_t)}} + \epsilon_t\sqrt{\frac{(1-\epsilon_t)}{\epsilon_t}}$$

[by using value of α_t]

$$\Rightarrow \boxed{Z_t^{opt} = 2\sqrt{\epsilon_t(1-\epsilon_t)}}$$

(Hence proved) ∴ (Answer)

(P.T.O)

1)

3.b) Given, $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$ and

$$\epsilon_t = \frac{1}{2} - \gamma_t$$

Therefore,

$$Z_t = 2\sqrt{\left(\frac{1}{2}-\gamma_t\right)\left[1-\left(\frac{1}{2}-\gamma_t\right)\right]}$$

$$= 2\sqrt{\frac{1}{4} - \gamma_t^2}$$

$$= \sqrt{1-(2\gamma_t)^2}$$

by taking log on both sides, we get,

$$\ln(Z_t) = \frac{1}{2}\ln(1-(2\gamma_t)^2) \quad \text{——— (1)}$$

Now, we know that,

Z_t is a real number

and $\gamma_t > 0$; which implies that

$$0 < \gamma_t \leq \frac{1}{2}$$

Also, we know the fact that

$$\log(1-x) \leq -x \text{ for } 0 \leq x \leq 1$$

So, equation - (1) can be re-written as:

$$2\ln(Z_t) = \ln(1-(2\gamma_t)^2) \leq -(2\gamma_t)^2$$

$$\Rightarrow 2\ln(Z_t) \leq -(2\gamma_t)^2$$

$$\Rightarrow \ln(Z_t) \leq -2\gamma_t^2$$

$$\Rightarrow \boxed{Z_t \leq e^{-2\gamma_t^2}}$$

(Hence proved)

1)

3.e)

Given,

$$\text{Etraining} \leq \prod_{t=1}^{T} Z_t \leq e^{-2\sum_{t=1}^{T}\gamma_t^2}$$

and, $\gamma_t \geq \gamma$

therefore,

$$\gamma_t^2 \geq \gamma^2$$

$$\Rightarrow \sum_{t=1}^{T}\gamma_t^2 \geq T\gamma^2$$

$$\Rightarrow -2\sum_{t=1}^{T}\gamma_t^2 \leq -2T\gamma^2$$

$$\Rightarrow e^{-2\sum_{t=1}^{T}\gamma_t^2} \leq e^{-2T\gamma^2}$$

(P.T.O)

Now, from given information,

$$\epsilon_{training} \leq \prod_{t=1}^{T} Z_t \leq e^{-2\sum_{t=1}^{T} \gamma_t^2} \leq e^{-2T\gamma^2}$$

$$\Rightarrow \boxed{\epsilon_{training} \leq e^{-2T\gamma^2}}$$

(Hence proved)

2.

1st To Do:

Define model for training on image: implemented in Action_CNN.ipynb

2nd To Do:

Define optimizer: implemented in Action_CNN.ipynb

3rd To Do:

Train the model to predict on image data set: implemented in Action_CNN.ipynb

4th To Do:

Steps followed:

1.I have constructed the model with the layer sequence like:
conv2d->batchNorm2d->relu->conv2d-batchNorm2d->relu->maxpool2d-> conv2d->batchNorm2d->relu->conv2d-batchNorm2d->relu->maxpool2d ->flatten->relu->linear

- conv2d : with 8 filters of size 3*3 and stride 1
- batchnorm2d with argument of 8
- conv2d : with 16 filters with size 3*3 and stride 1
- batchnorm2d with argument of 16
- maxpool2d : with kernel size 2 and stride 2
- conv2d : with 32 filters of size 3*3 and stride 1
- batchnorm2d with argument of 32
- conv2d : with 32 filters with size 3*3 and stride 1
- batchnorm2d with argument of 32
- maxpool2d : with kernel size 2 and stride 2
- linear : with size 5408 and with 10 classes

2.I have created the model base with the defined model structure

3.Defined CrossEntropyLoss as loss function

4.Tried the below optimizer functions with the defined model:

a) Adadelta optimizer which was not providing much good result but this was training the network very slowly

b) ASGD optimizer accuracy on the validation set was not much

c) Adam optimizer which gave some satisfactory result

5.Then I trained my model with Adam and with 10 epochs with validation accuracy : 60.18

6.Predicted action values on the test dataset using the trained model and got count 9810

7.Uploaded the result ‘results.csv’ in Kaggle with the name of 'gourabbhattacharyya' and got the accuracy of 63.25178

5th To Do:

Test model on test image dataset and submit the result on Kaggle: implemented in Action_CNN.ipynb and submitted 'results.csv' on Kaggle

6th To Do:

Flatten the 3d convolution feature maps: implemented in Action_CNN.ipynb

7th To Do:

Define model for 3d convolution video classification: implemented in Action_CNN.ipynb

8th To Do:

Steps followed:

1.I have constructed the model with the layer sequence like:

conv3d->batchNorm3d->dropout3d->relu->maxpool3d->flatten->relu->linear

- conv3d : 32 filters with 3*3 size, stride (1, 2, 2) and padding (3, 3, 3)
- batchNorm3d : with 32 values
- dropout3d : with 0.15 probability
- maxpool3d : with size (3, 3, 3)
- linear : with size 36992 and 10 classes

2.I have created the model base with the defined model structure

3.Defined CrossEntropyLoss as loss function

4.Tried the below optimizer functions with the defined model:

a) Adadelta optimizer but this was training the network very slowly and accuracy on the validation set was not much

b) ASGD optimizer which was again not providing much good result

c) RMSprop optimizer which gave some satisfactory result

5.Then I trained my model with RMS prop and with 12 epochs with validation accuracy : 64.93

6.Predicted action values on the test dataset using the trained model and got count 3270

7.Uploaded the result ‘results_3d.csv’ in Kaggle with the name of 'gourabbhattacharyya' and got the accuracy of 64.03669