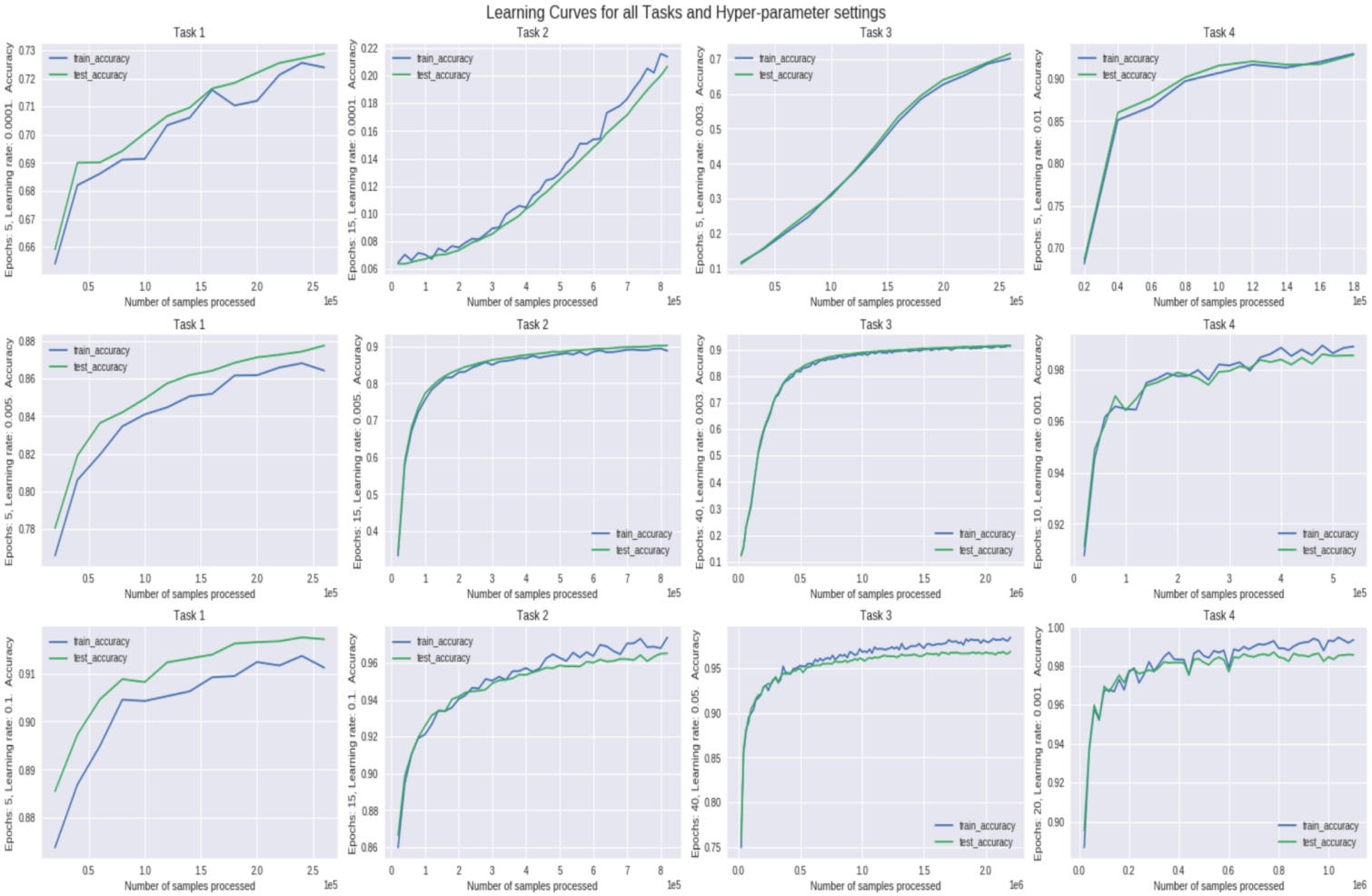
Deep Learning Coursework 2

WU Zhi 17040772 UCABWUA@UCL.AC.UK
Department of Computer Science
University College London

Plot



Result

	Setting 1	Setting 2	Setting 3
Model 1	0.7288	0.8776	0.9171
Model 2	0.2066	0.9026	0.9651
Model 3	0.7146	0.9148	0.9685
Model 4	0.9281	0.9857	0.9856

Question

Q1

(1)

Beacuse

$$loss = -\sum_{i=1}^{N} y_{label} \log\left(Softmax(z)\right)$$

So

$$\frac{\partial loss}{\partial z} = \frac{\partial loss}{\partial Softmax} \cdot \frac{\partial Softmax}{\partial z}$$

And

$$Softmax = \frac{\exp(z[i])}{\sum_{c=1}^{10} \exp(z[c])}$$

So calculate $\frac{\partial Softmax}{\partial [j]}$, two conditions need to be considered:

- j is equal to i

$$\frac{\partial Softmax}{\partial z[j]} = \frac{\exp(z[j]).\sum_{c=1}^{10}\exp(z[c]) - exp(z[j]).exp(z[j])}{(\sum_{c=1}^{10}\exp(z[c]))^2} = \frac{exp(z[j])}{\sum_{c=1}^{10}\exp(z[c])} - \frac{exp(z[j])}{\sum_{c=1}^{10}\exp(z[c])} \cdot \frac{exp(z[j])}{\sum_{c=1}^{10}\exp(z[c])}$$
$$= Softmax(z[j])(1 - Softmax(z[j]))$$

- j is not equal to i

$$\frac{\partial Softmax}{\partial z[j]} = \frac{0.\sum_{c=1}^{10}\exp(z[c]) - exp(z[j]).exp(z[i])}{(\sum_{c=1}^{10}\exp(z[c]))^2} = \frac{exp(z[j])}{\sum_{c=1}^{10}\exp(z[c])} \cdot \frac{exp(z[i])}{\sum_{c=1}^{10}\exp(z[c])} = -Softmax(z[j])Softmax(z[i])$$

$$\frac{\partial loss}{\partial z} = -\sum_{i=1}^{N} \frac{\partial y_{label}\log\left(Softmax(z)\right)}{\partial Softmax} \cdot \frac{\partial Softmax}{\partial z} = -\sum_{i=1}^{N} y_{label} \frac{1}{Softmax} \cdot \frac{\partial Softmax}{\partial z}$$

$$\frac{\partial loss}{\partial z[i]} = -y_{label}[i](1 - Softmax(Z[i])) - \sum_{c=/=i}^{10} y_{label}[c] \frac{1}{Softmax(z[c])} \cdot (-Softmax(z[c]).Softmax(z[i]))$$

$$= -y_{label}[i](1 - Softmax(Z[i])) + \sum_{c=/=i}^{10} y_{label}[c] Softmax(z[i]) = Softmax(z[i])(y_{label}[i] + \sum_{c=/=i}^{10} y_{label}[c]) - y_{label}[i]$$

Because y is one hot encoded vector for the labels, so $\sum_{c}^{10} y_{label}[c] = 1$, and $y_{label}[i] + \sum_{c=/=i}^{10} y_{label}[c] = 1$. So we have:

$$\frac{\partial loss}{\partial z[i]} = Softmax(z[i]) - y_{label}[i]$$

(2)

Because $\frac{\partial loss}{\partial x} = \frac{\partial loss}{\partial z}\frac{\partial z}{\partial x}$, and $\frac{\partial z}{\partial x} = w$, so:

$$\frac{\partial loss}{\partial x} = \frac{\partial loss}{\partial z}\frac{\partial z}{\partial x} = (Softmax(z[i]) - y_{label}[i]).\, w$$

Because $\frac{\partial loss}{\partial b} = \frac{\partial loss}{\partial z}\frac{\partial z}{\partial b}$, and $\frac{\partial z}{\partial b} = 1$, so:

$$\frac{\partial loss}{\partial b} = \frac{\partial loss}{\partial z}\frac{\partial z}{\partial b} = Softmax(z[i]) - y_{label}[i]$$

Because $\frac{\partial loss}{\partial w} = \frac{\partial loss}{\partial z}\frac{\partial z}{\partial w}$, and $\frac{\partial z}{\partial w} = x$, so:

$$\frac{\partial loss}{\partial w} = \frac{\partial loss}{\partial z}\frac{\partial z}{\partial w} = (Softmax(z[i]) - y_{label}[i]).\, x$$

(3)

In the process of forward propagation convolution layer, $z_{ij}^k = (w^k * x)_{ij}$ i and j are the position, and k is the depth of the filter. x is a 4 dim input, and w is a $H * W * D$filter.

$$\frac{\partial loss}{\partial W} = \frac{\partial loss}{\partial Conv}\frac{\partial Conv}{\partial w}$$

$$\frac{\partial loss}{\partial W_{H,W}^D} = \sum_{u,v}(\frac{\partial loss}{\partial Conv}^D{}_{H,W})_{u,v}(x_{H,W}^D)_{u,v}$$

$x_{H,W}^D$ is a $H * W$ patch, and the center pisition of this patch in input is (u, v)

Q2

(1)

Model 1: The running time for model 1 is ok, but the all model-1 3 settings results have the lowest accuracy, and this is because linear model is too simple for training. Hence, model 1 result is unstable.

Model 2: The running time for model 2 is ok, but the model-2 first setting results have the low training and test accuracy, and this is because the learning rate(0.0001) is too small.

Model 3: The running time for model 3 is much more than model 1 and 2, because the network is more complicate than previous model. This my code, i use many loops in forward propagation and backward propagation, and this speed more time for program to train the models. The model-3 result is very stable. The setting 1 result is lower than setting 2 and 3, and this is because the number of epochs is too small.

Model 4: The running time for model 4 is around 10 hours, and i speed all the night to finish training. The result is good, both test and training accuracy are good. Hence the result is not very stable.

(2)

In the experiments above, model 2 first setting (num_epochs=15, learning_rate=0.0001) are examples of under-fitting. This is because the netowrk is too simple for training data.

(3)

In the experiments above, model 2， 3 and 4 third setting are examples of over-fitting. This is because the netowrk is too complex for training data.