AI & Deep Learning 2023

1. (20%) (Optimization)

   (a) (10%) Explain the SGD (Stochastic Gradient **Descent**) updating policy. You need to explain SGD in terms of gradient of loss function $\nabla(J(w))$, parameters (**w**), and learning rate ε. (10%). (Note that a '+' or a '-' sign in the equation have different meaning. Don't mix them up!)

   (b) Since SGD may be stuck in a local optimum or a saddle point, how can SGD be improved? (hint: momentum—accumulated gradient; adaptive subgradient)

2. (20%)(CNN)

   (a) (10%) Please compute the output feature map of input **A** after it is convolved with filter **W** using **stride of 1 and "same padding."** Assume that bias b = 0.

1	1	0	1
2	0	1	2
2	0	2	1
1	2	0	2

W

1	0	0
0	1	0
0	0	1

   (b) (10%) Perform a 2*2 max-pooling (with **stride of 2**) on the feature map derived in question (a). Draw the resultant feature map.

3. (20%) The following code segment is a Keras code for an MLP neural network. The input is a one-dimensional array of 784 elements (a hand-writing decimal digit).

```
In [7]: from keras.models import Sequential
        from keras.layers import Dense

        Using TensorFlow backend.

In [8]: model = Sequential()

In [9]: model.add(Dense(units=256,
                        input_dim=784,
                        kernel_initializer='normal',
                        activation='relu'))

In [10]: model.add(Dense(units=10,
                         kernel_initializer='normal',
                         activation='softmax'))
```

(a) (10%) What is the total number of parameters in **the first dense layer** of the mlp?

(b) (10%) What is the total number of parameters in **the second dense layer** of the mlp?

4. (Cross Entropy) (20%) Calculate the cross entropies of the prediction1 and prediction2 in the following using the cross entropy equation of H. Based on the cross entropies, which prediction is more accurate? (Note that in H: c is number of classes, n is number of examples, $y_{c,i}$ is the one-hot encoding value of class c for example i, and $p_{c,i}$ is the predicted value of class c for example i.)

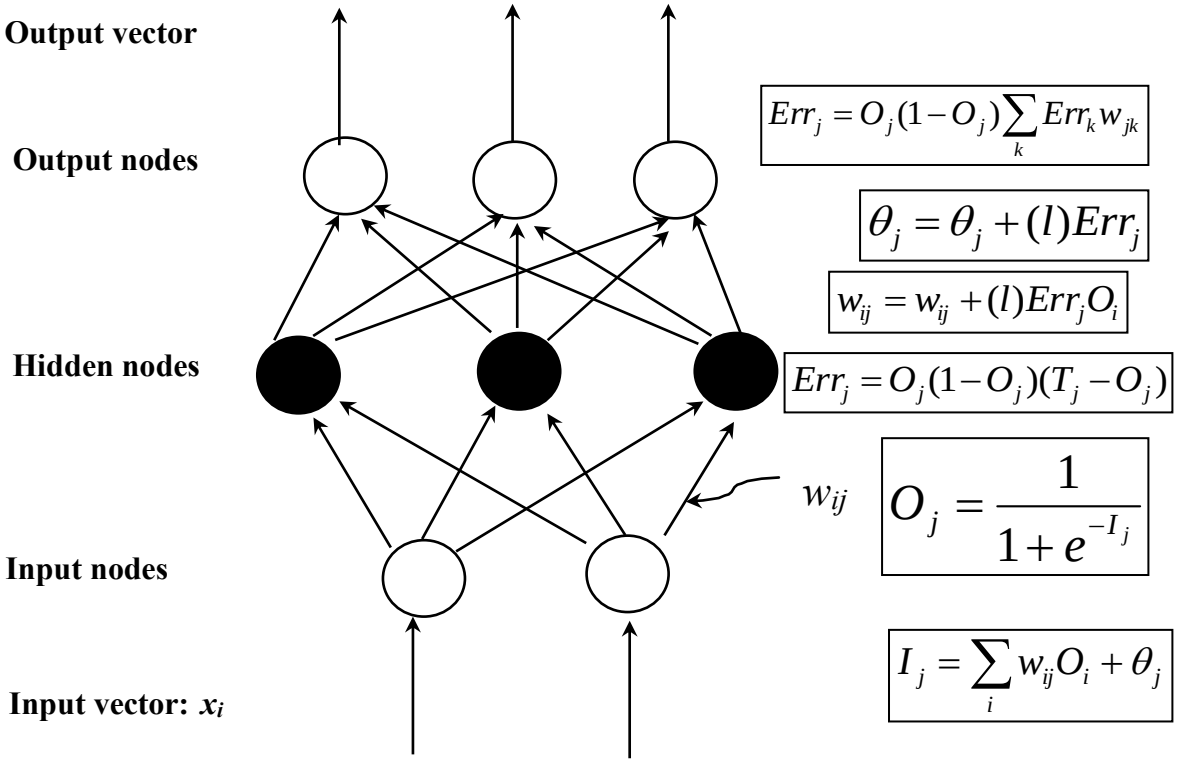
$$H = \sum_{c=1}^{C} \sum_{i=1}^{n} -y_{c,i} log_2(p_{c,i})$$

Prediction 1							
		Predicted prob.			Actual class (one hot encoding)		
		boy	girl	other	boy	girl	other
data1	boy	0.4	0.3	0.3	1	0	0
data2	girl	0.3	0.4	0.3	0	1	0
data3	boy	0.5	0.2	0.3	1	0	0
data4	other	0.8	0.1	0.1	0	0	1

Prediction 2							
		Predicted prob.			Actual class (one hot encoding)		
		boy	girl	other	boy	girl	other
data1	boy	0.7	0.1	0.2	1	0	0
data2	girl	0.1	0.8	0.1	0	1	0
data3	boy	0.9	0.1	0.0	1	0	0
data4	other	0.4	0.3	0.3	0	0	1

5. (20%) (MLP) Consider a simple MLP with two inputs, a and b, one hidden node c, and one output node d. Please find the updated weights of $\mathbf{W_{cd}}$ and $\mathbf{W_{ac}}$ after training the network once with the following sample, where a and b are input values, and d is the corresponding output value of the network. Assume that the initial values of weights w_{ac}, w_{bc}, w_{cd} are 0.1, 0.1, 0.1, respectively, and the learning rate is 0.1.

a	b	d
1	1	1



(notes:1. $\mathbf{W_{cd}}$ denotes the weight of the link between node **c** and node **d)**