**=1=**

**Exercise sheet 1**

**1** An output model is obtained by learning the characteristics of the sample, and a prediction result can be output by inputting an unknown instance into the model. Speech Recognition, Image classification,Processing loan application, Diagnosis of machine faults, Electricity supply forecasting.

**2** Classification: In samples of known class, the model learns features of different class. Inputting an unknown class instance into the model and then can predict the class of the this instance. Clustering: In samples of unknown class, the model divides samples with similar features into different classes. Generally, classification belongs to supervised learning, and clustering belongs to unsupervised learning.

**3** **R**egression model learning the relationship between the feature variables and the objective function. Input new feature variables into the regression model, and the model can predict a target value. The output of the classification model is a discrete value, while the output of the regression model is a continuous value.

**4** **Corona-19** infection problem: the dependent class is infection[0] and no infection[1], the attributes are [0]Whether you have a cough, [1]whether you have a fever, [2]whether you are weak. Housing price prediction problem: the dependent target is price of house, the attributes are [0]size of hose, [1]location of house, [2]age of house.

**5** Customer cluster on bank loan issues. I will cluster customers based on the attributes of [0] income, [1] fixed assets, [2] education, and [3] family status. I will try to use K-means to divide customers into three categories, high-quality customers, ordinary customers and high-risk customers.

●Classification (supervised)

Predicting class membership. Predicting a discrete value

● Numeric Prediction (regression, supervised)

Predicting a numeric value

● Association

Determining associations between arbitrary features

● Clustering (unsupervised)

Grouping of objects based on their similarity

**2**

**Exercise 1: attribute type**

Nominal: the attribute can be described by Nominal value. No relation is implied among nominal values (no ordering or distance measure)

Ordinal：the attribute values is in order. But no distance between values defined

Interval: the attribute values are not only ordered but also measured in fixed and equal unit.

Difference of two values makes sense. Sum or product doesn’t make sense. Because zero point is not defined!

Ratio: the attribute value are measured from zero point

**Exercise 2: attribute type**

Nominal: Such as the attribute “Outlook”, the value can be rain, sun and so on.

Ordinal： Such as attribute “Temperature”, the value can be hot , middle, cool, and hot > middle > cool.

Interval :Such as attribute “Year”, the value can be 1 year, 2 year, ... and so on.

Ratio: Such as attribute “distance”, the value can be 0m, 2m and so on.

**Exercise 3:** **missing value ( 3 11)**

I need to know if the missing value have some significance. If yes, “missing” is a separate value. If no, “missing” must be treated in a special way.

I will make the earnings missing value = the average of this attribute values of all instances. For gender attribute missing value, i will make the missing value = the most popular value of this attribute in all instances.

**Exercise 4: Normalization**

Data normalization means that the data will be mapped to a certain interval. In different samples, one attribute may have some different scale, such as the height of a person, the scale can be in cm our m. so we can map all this attribute values in to range [0, 1]. We perform data normalization in the data pre processing stage, generally before the training stage.

Due to different attributes are measured on different scales, so we need to be normalized

**Exercise 5: RapidMiner**

The allowed data types are: real, integer, nominal and binominal. RapidMiner Radoop stores real and integer attributes in Hive as DOUBLE and BIGINT columns; nominal attributes are stored as STRING columns; binominal attributes are stored as either STRING or BOOLEAN columns.

3

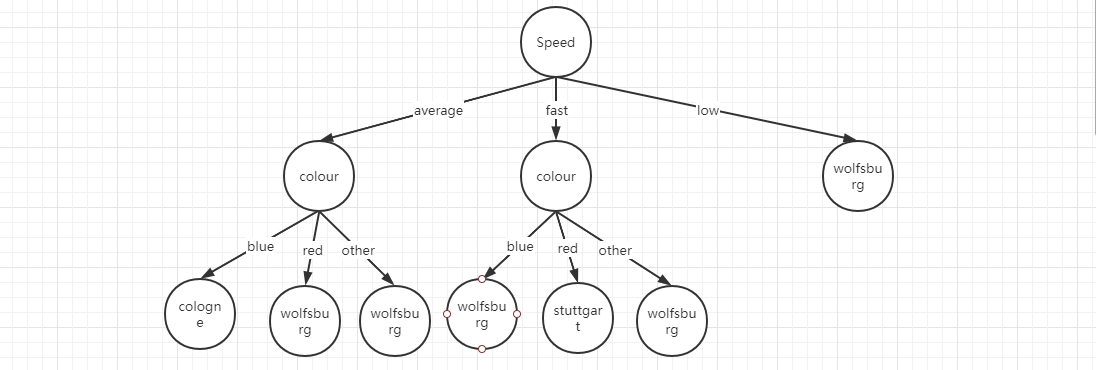
**Exercise 1: Naive Bayes (4 14)**

P(H|E) = P(E|H)\*P(H)/P(E). Prior probability is probability of event before evidence is seen. Statistical independence: evidence splits into some attributes that are conditionally independent. For classification problem, the evidence is instance’s non-class attribute values and the event is the class of the instance.

**Exercise 2: Transformation of rules to decision trees**

(a) Create a decision tree based on these rules.

The attribute speed can have the values fast, average, slow. The attribute colour consists of the values red, blue or other.



(b) How are decision trees transformed to rules?

Antecedent contains a condition for every node from the root to the leaf, and the result is the class assigned by leaf.

(c) Why is the transformation of rules to decision trees considered more complex than the transformation of decision trees to rules?

The tree cannot easily express disjunction between rules.

**Exercise 3: Rules and decision trees in *RapidMiner***

(a) Create two processes, which will create a decision tree and rules based on the given data. The attribute label is the class.

*Hint*: The creation of the decision tree and the induction of the rules may

take several minutes on slower PCs.

**4**

**Exercise 1: overfitting**

Overfitting: The model performs well on training data, but does not perform well on generalization

Causes: The model learns the detail and noise in the training data. Solutions: A. Reduce the complexity of the model, B. Increase the diversity of samples

C: reduce noise of training data. D: collect more training data.

Underfitting occurs when the model has not trained for enough time or the input variables are not significant enough to determine a meaningful relationship between the input and output variables.

**Exercise 2: Linear Regression**

We will use the following training set of a small sample of different students’ performances:

1. 1/2
2. 4

**Exercise 3: Numeric prediction with *RapidMiner***

1.\*

1. Y = W0 + W1 \*MYCT+W2\*MMIN +W3\*MMAX+W4\*CACH+W5\*CHMIN+W6\*CHMAX
2. 205.244

**Exercise 4: Naive Bayes**

With the *Naive Bayes* function the probability of *H* (hypothesis or event) given an *E* (evidence) is calculated:

(a) Calculate the probability *P r*(*H|E*) for both instances.

Use the modified probability estimates (as seen in the lecture using the Laplace estimation) with *µ* = 1 (Attribute outlook), if a problem with 0 frequency occurs.

P(E) = 5/14 \* 4/14 \* 7/14 \* 8/14

P(E|H) = P(E1|H)P(E2|H)P(E3|H)P(E4|H) = 3/9 \* 2/9 \* 6/9 \* 6/9

P(H) = 9/14

P(H|E) = 72.5%

P(E) = 5/14 \* 4/14 \* 7/14 \* 8/14

P(E|H’) = P(E1|H’)P(E2|H’)P(E3|H’)P(E4|H’) = 2/5 \* 2/5 \* 1/5 \* 2/5

P(H’) = 5/14

P(H’|E) = 15.7%

Yes

P(E) = 4/14 \* 4/14 \* 7/14 \* 6/14

P(E|H) = P(E1|H)P(E2|H)P(E3|H)P(E4|H) = 4/9 \* 2/9 \* 6/9 \* 3/9

P(H) = 9/14

P(H|E) = 80.6%

P(E) = 4/14 \* 4/14 \* 7/14 \* 6/14

P(E|H’) = P(E1|H’)P(E2|H’)P(E3|H’)P(E4|H’) = (1/3)/(5+1) \* 2/5 \* 1/5 \* 3/5

P(H’) = 5/14

P(H’|E) = 5.4%

Yes

5

**Exercise 1: Logistic Regression**

P(y = 1|x; w) = 0.4, P(y = 0|x; w) = 0.6

**Exercise 2: High variance and High Bias**

High variance means the output of the model is distributed around the true value, as shown in the following figure. In other words, the model over fits the training set, so its performance on the test set is unstable. The output of the model is far from the true value, such as shooting a target. The model's under fitting on the training set resulted in not learning enough features. High Bias - Low Variance (Underfitting): Predictions are consistent, but inaccurate on average. This can happen when the model uses very few parameters.

High Bias - High Variance: Predictions are inconsistent and inaccurate on average.

Low Bias - Low Variance: It is an ideal model. But, we cannot achieve this.

Low Bias - High Variance (Overfitting): Predictions are inconsistent and accurate on average. This can happen when the model uses a large number of parameters.

What is Bias?

Bias is the difference between the average prediction and the correct value. It is also known as Bias Error or Error due to Bias.

Low Bias models: k-Nearest Neighbors (k=1), Decision Trees and Support Vector Machines.

High Bias models: Linear Regression and Logistic Regression.

What is Variance?

Variance is the amount that the prediction will change if different training data sets were used. It measures how scattered (inconsistent) are the predicted values from the correct value due to different training data sets. It is also known as Variance Error or Error due to Variance.

Low Variance models: Linear Regression and Logistic Regression.

High Variance models: k-Nearest Neighbors (k=1), Decision Trees and Support Vector Machines. Overfitting: It is a Low Bias and High Variance model. Generally, Decision trees are prone to Overfitting.

Underfitting: It is a High Bias and Low Variance model. Generally, Linear and Logistic regressions are prone to Underfitting.

How to identify High Variance or High Bias?

Identifying High Variance / High Bias

High Variance can be identified when we have:

Low training error (lower than acceptable test error)

High test error (higher than acceptable test error)

High Bias can be identified when we have:

High training error (higher than acceptable test error)

Test error is almost same as training error

How to address High Variance or High Bias?

High Variance is due to a model that tries to fit most of the training dataset points making it complex. Consider the following to reduce High Variance:

Reduce input features(because you are overfitting)

Use less complex model

Include more training data

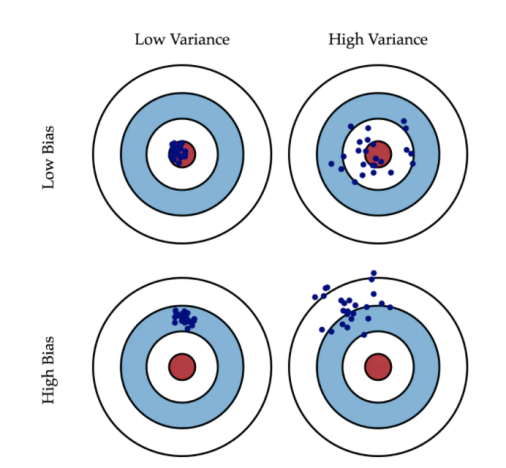
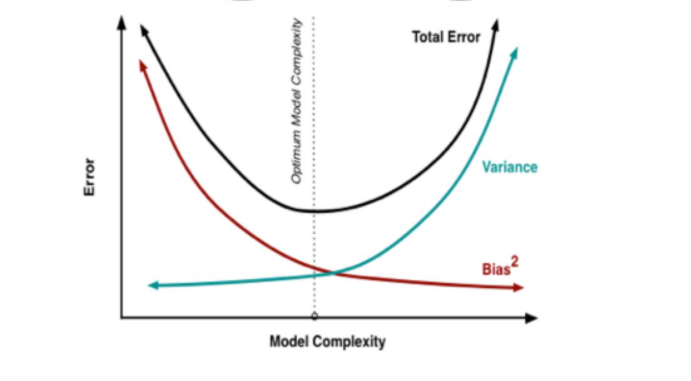
Increase Regularization term

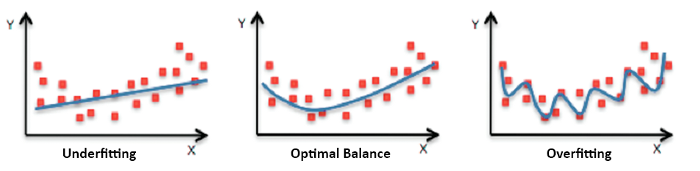
High Bias is due to a simple model. Consider the following to reduce High Bias:

Use more complex model (Ex: add polynomial features)

Increase input features

Decrease Regularization term



**Exercise 3: KNN**

The smaller the K, the more complex the model is, over-fitting may occur, and the prediction results are more sensitive to neighboring points (maybe noise points). The larger K may lead to relatively large prediction errors. I may try to increase the value of K, and I will use cross-validation to select an appropriate value of K.

**Exercise 4: K-means clustering**

Step 1 : Choose k random cluster centers

Step 2 : Assign each instance to its closet cluster center based on Euclidean distance.

Step 3 : Recompute cluster centers by computing the average(aka centroid) of the instances pertaining to each cluster.

Step 4 : This algorithm minimizes the squared Euclidean distance of the instance from their corresponding cluster centers.

Step 5 : if Cluster center have moved, go back to step 2, else stop.

The size of the seed determines the number of classes. The elbow method can be used to determine the K value, increase the K value, calculate the distance from all points of the cluster under the K value to the center point, and then sum it as the SSE(Sum of Squared Error). The K point where the WSS (within-cluster sum of square) is significantly reduced is the best K value.

**Exercise 5: Evaluation methods**

I give a example about email classification, normal email is Positive[1] and the spam email is Negative[0].

TP: the prediction result is normal email and this email’s true label also is normal email

TN:the prediction result is spam email and this email’s true label also is spam email.

FP:the prediction result is normal email but this email’s true label is spam email.

FN:the prediction result is spam email but his email’s true label is normal email.

Accuracy : TP+TN/TP+TN+FP+FN

Mean-squared error: The sum of the squares of the difference between the predicted value and the true value of all samples.

Precision:TP/TP+FP

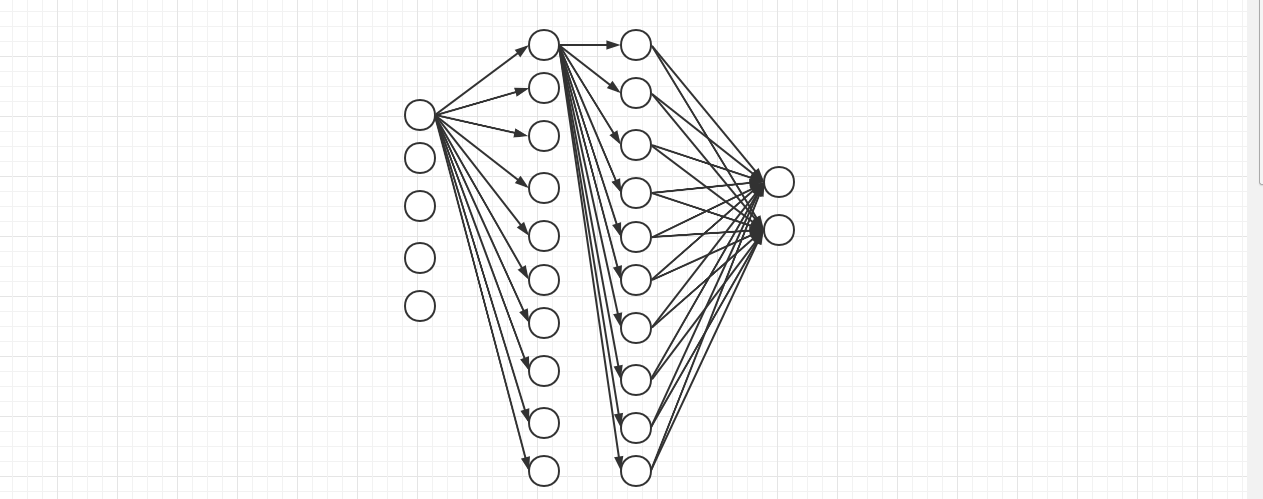
Recall : TP/TP+FN

F1=2P\*R/P+R

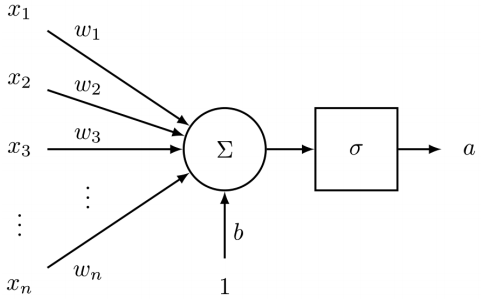
(c) Mean-squared error

7

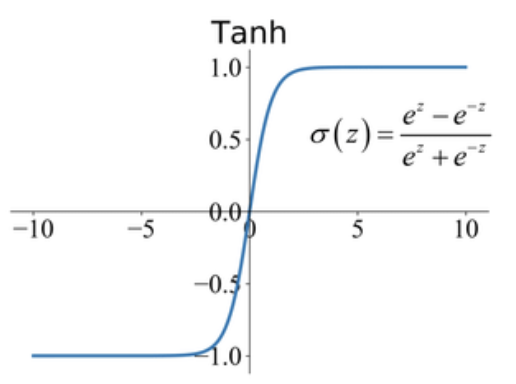
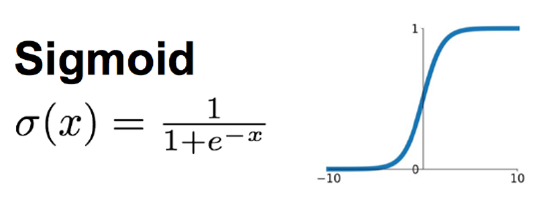
**Exercise 1: Neural Networks**

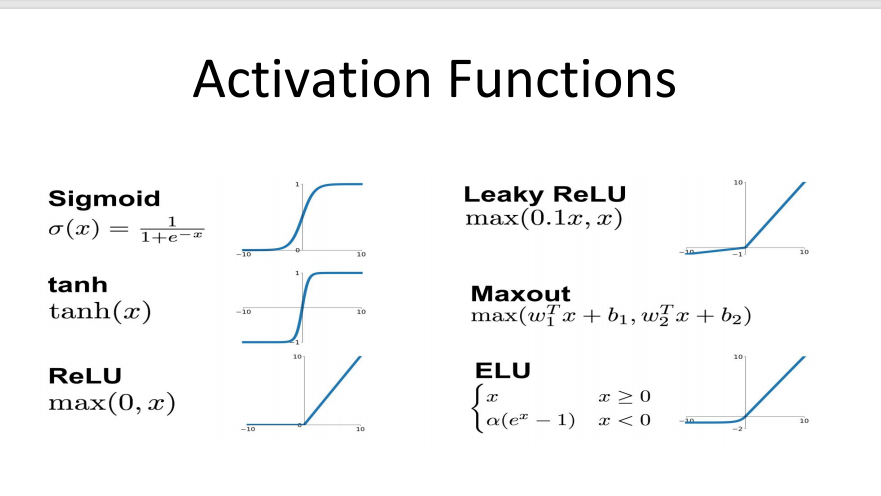


**Exercise 2: Neuron architecture**



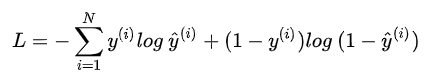
**Exercise 3: Activation functions**

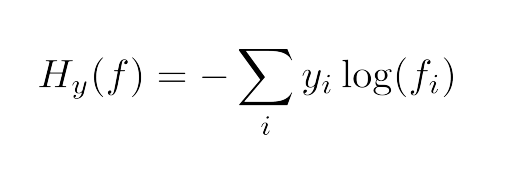


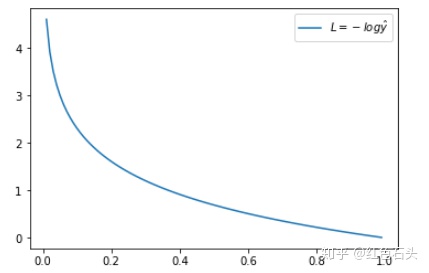
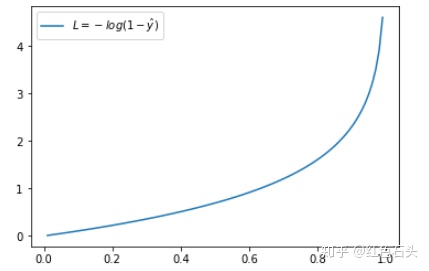


The activation function introduces a nonlinear factor to the model.

**Exercise 4: Cross entropy**

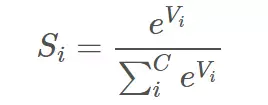


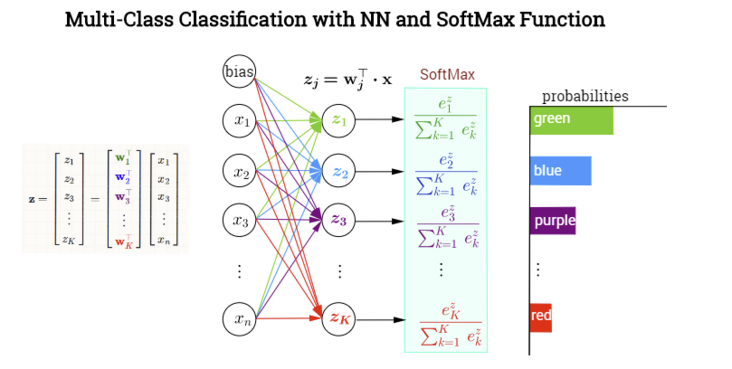


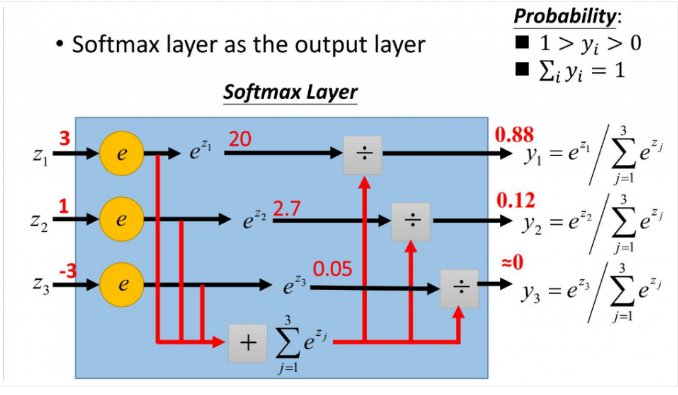
 

**Exercise 5: Softmax**

**The softmax function helps us transform these output values into probability distributions:** each output can be treated as the probability of that class.





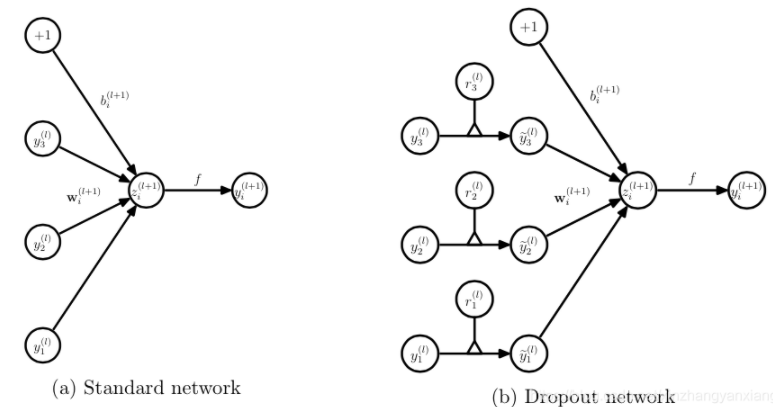


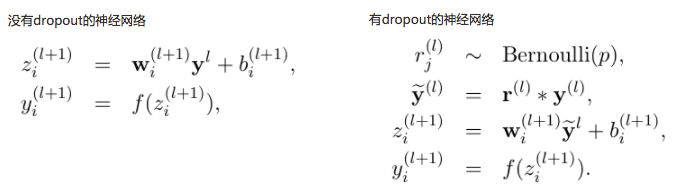
8

**Exercise 1: Dropout**

“Randomly drop neurons from the network during training”

In order to avoid overfitting the training data and make the training of the model more robust and improve generalization.

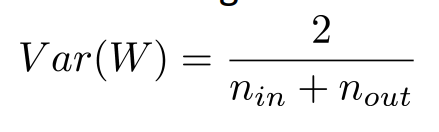




**Exercise 2: Weight initialization**

Set some initial values for the parameters in the neural network. 1. Random initialization (Avoids symmetry in the network by having different random weights)2. Xavier initialization(pick the

random numbers from a distribution with zero mean and the following variance)



3. He initialization. This is because if w is initialized to 0, All the neurons see the same input -and with the same weight matrices, they will make the exact same decisions

**Exercise 3: Hyper parameters**

In the context of machine learning, hyper parameters are parameters whose values are set before starting the learning process, not parameter data obtained through training. Such as the number of layers of the neural network, the learning rate, the K value of the cluster, etc.

**Exercise 4:** **Embedding**

In a matrix, if the number of elements with a value of 0 is far more than the number of non-zero elements, and the distribution of non-zero elements is irregular, the matrix is called a sparse matrix; on the contrary, if the number of non-zero elements is the majority , The matrix is called a dense matrix. An embedding is a relatively low-dimensional space into which you can translate high-dimensional vectors. Embedding make it easier to do machine learning on large inputs like sparse vectors representing words. Ideally, an embedding captures some of the semantics of the input by placing semantically similar inputs close together in the embedding space. An embedding can be learned and reused across models.

word vectors are continuous representations of words. vectors of different words give us information about the potential relations between the words - words closer together in meaning have vectors closer to each other.

The benefits:

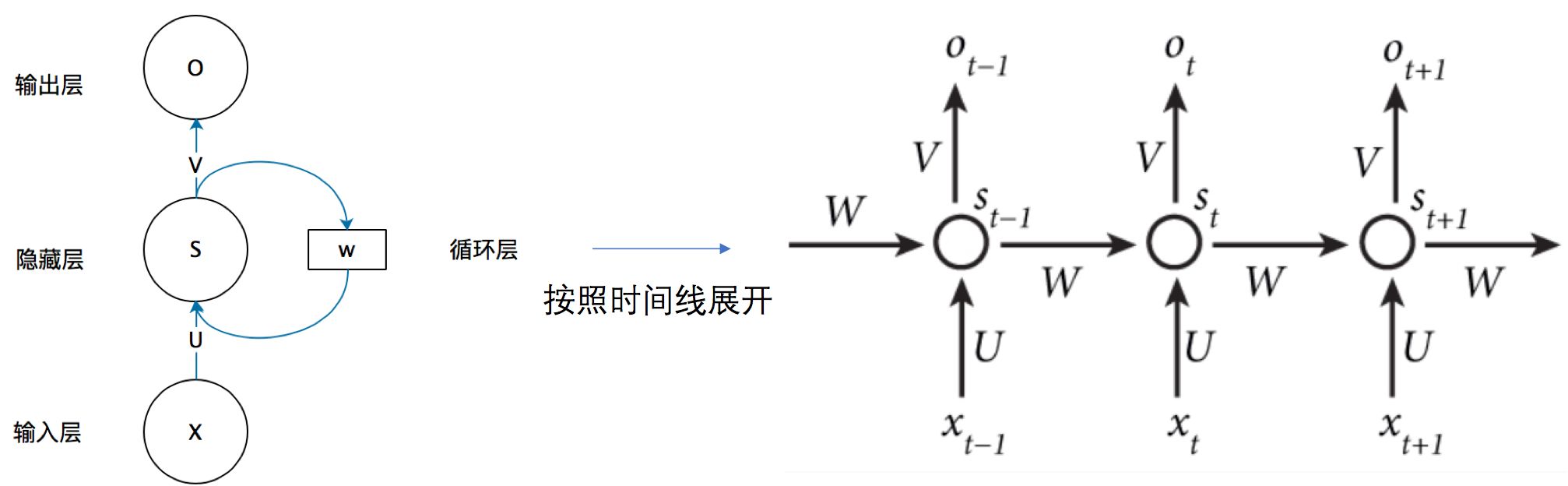
Reduce dimensionality

Semantic relatedness

Increase expressiveness

**Exercise 5: RNN**

RNN is mainly used for sequence data processing, such as text translation, which has a high degree of correlation between input and output sequences.



**Exercise 6:** **LSTM**

LSTM can avoid the gradient vanishing of RNN. In order to remember the long-term state, LSTM adds one input and one output on the basis of RNN. The added path is the cell state, which is the top path on the way.

We have a “memory cell” or “cell state” that is passed along the time steps. At each timestep, the unit

decides to forget some information from this cell and add some new information from the current input!

This effectively helps us solve both the Information decay and the vanishing gradient problem

**Exercise 7: Deeper understanding questions**

No

D

**Exercise 8: CNN basics**

Compared with fully connected neural networks, CNN can capture the local features of pictures/data, prevent over fitting, and better simulate the human brain.

The convolution kernel is a parameter matrix, which allows different convolution kernels and pictures to perform convolution operations to extract different features.

After a convolution kernel performs sliding convolution on the one layer of input image, a new map is obtained, called an activation map.

When doing sliding convolution operation, the length of each movement of the convolution kernel.

**Exercise 9: CNN Padding**

After the convolution operation, in order to keep the information of the feature map not lost. **Full padding** indicates that the convolution operation is performed when the convolution kernel and the image begin to intersect and the size of the image generated by the convolution is larger than the original image. **Same padding** indicates that the convolution operation is performed when the convolution kernel intersects the center of the image and the size of the image generated by convolution is the same as the original image. **Valid padding** indicates that the convolution operation is performed when the convolution kernel is completely in the image and the size of the image generated by the convolution is smaller than the original image. The free part is filled with some values, generally 0.

5 x 5, 3 x 3

**Exercise 9: CNN Filter**

When processing image data, the width and height of the convolution kernel used by CNN are the same, but in text-CNN, the width of the convolution kernel is consistent with the dimension of the word vector. 22 x 22

**Exercise 10: Pooling**

Reduce the dimension of the information extracted by the convolution layer and reduce the amount of calculation, and strengthen the invariance of image features. average-pooling, max-pooling. 9