note for midterm and finalexam

L1

preface: thoughts about AI

AI Latest Development : AI in Creative Arts Last Week

Google's AI development: ALPHAFOLD 2022 (attention but not CNN)

初代从局部开始预测，忽略长距离依赖性，同时聚焦多个细节部分

多序列比对（MSA）

均使用attention 神经网络（图网络和MSA结合）和结构模块

Elon Musk and AI

From Bot to Our Symbiosis with AI 共生

brain and computer

China AI advancement

AI innovation center

Next for AI from award winners

九章 Quantum Computing

L2

My Research Area – Medical AI

Eye-brain joint research

Precision Medicine

Artificial Intelligent

Computer-assisted surgery

My Main Research Area – Ocular AI

Group projects and surveys

L3

AI Concepts – "Intelligence" from Dictionaries

The ability to **Learn Understand Deal with Try** new situations

It is the science and engineering of making intelligent machines, especially intelligent computer program

AI Categories(Q1)

Symbolicism 符号主义 （Logicism (逻辑主义)，Psychologism (心理学派), Computerism (计算机学派))

Major Components

• Mathematical Logic

• Expert System

• Knowledge Engineering

• Intelligence are assigned by **Human**

Connectionism

Connectionism (联接主义或连接主义) are also called

• Bionicsism (仿生学派)

• Physiologism (生理学派)

Major Components

• Artificial Neuron

• Perceptrons

• BP Neural Networks

• DNN

• Intelligence are learned by **Machine**

Actionism

• Actionism (行为主义) are also called

• Evolutionism (进化主义)

• Cyberneticism (控制论学派)

• Major Components

• Reinforcement Learning

• Robotics

• Intelligence are learned from the **feedback of the environment**

In Real Life, AI is often the **combination** of the 3

AI from Computer Science

Artificial intelligence (AI) is the branch of computer science that **develops machines and software with intelligence**

Major AI researchers and textbooks define the field as "**the study and design of intelligent agents**", where an intelligent agent is **a system that perceives its environment and takes actions that maximize its chances of success**

Agent

perceive its environment through sensors

acts upon that environment through effectors (actuators)

Abstractly, an agent is a function from percept histories to actions:

key components

PEAS?

1.sensors

2.effctors(actuators)

3.

Reinforcement Learning(Q2)

5 components: agent, state, action, reward, and environment

Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to **maximize the notion of cumulative reward**.

three basic machine learning paradigms:

supervised learning and unsupervised learning and reinforcement learning

DBS Agent deep brain simulation

Q Learning Algorithm

- Q-learning是一个value-based强化学习算法，用来通过q function查找最优的action-selection policy。
- 目标：使value function Q（expected future reward given a state and action）最大化。
- Function Q(state, action) → 返回在该state采取该action时的expected future reward。（对未来面临同样情况(state,action)时可能获得的reward的估计）
- 该function可以用Q-learning方法进行估计，Q-learning是用贝尔曼方程的方法对Q(s,a)迭代更新。

AI principle behind ALPHAGO :

1.Use two independent neural network, policy network and evaluation network.

Both are basically composed of 13-layer convolutional neural network, and the size of convolution kernel is 5*5

The policy network is basically a simple supervised learning of where the opponent is most likely to move.

The policy network uses reinforced-learning (RL) policy network and exclude some regions from convolution core before calculation to optimize.

The evaluation network calculates the win rate of each position given the current situation.

Use Monte Carlo search tree.

L4

AI and US - Future of AI?

narrow, general, super

AI Algorithm Summary

1950 AI

1980 ML

2010 DL

Machine Learning Algorithms

C4.5

classification and regression tree

naive Bayesian

SVM

KNN

AdaBooost

Kmeans

EM最大期望

DL algorithm

CNN

CNN imitates the process of human vision forming. Principle of convolutional neural network mimicking vision.

Principle of convolutional neural network:

(1) Principle of simulated vision: human visual recognition involves different levels of visual cortex, and each layer processes different transactions;

(2) **Construct multi-layer neural network model**: imitates the visual cortex, and creates multi-layer neural network model, such as convolutional neural network;

(3) **Hierarchical working mechanism**: multi-layer neural network model mechanism, the bottom layer recognizes the edge features of the image, the upper layer gradually recognizes the shape, and the upper layer determines and classifies the image pixels.

L5

• Pre-AI and Psychoanalysis

• Human Retina Vision and AI

• Layered Visual Pathway Network

• Human Brain Neurology and AI

• Electronic Brain

Computer Algorithm:

a well-defined sequence of steps for solving a computational problem

It produces the correct output

It uses basic steps / defined operations

It finishes in finite time

Vision and Neurology Research Directly Inspire AI

dominator-modulator

some optic nerve fibres (dominators) are sensitive to the whole spectrum while others (modulators) respond to a narrow band of light wavelengths and are thus color-specific.

光可以抑制 inhibit 也可以刺激simulate optic nerve

functional specialization of the cerebral hemispheres

information processing in the visual system.

 cells with similar eye preference were grouped together into columns, and eye dominance shifted periodically across the cortex

L6

Neocortex

The neocortex is sheet of neurons, a 1.7 mm thick and 1,300 cm2 in area, that constitutes the outermost part of the two cerebral hemispheres of the brain. • Every mm3 of neocortex contains 100,000 neurons, and the whole neocortex contains a total of approximately 25 billion neurons, each of which receives inputs via 4,000 synapses. • The global structure of the visual cortex is organized retinotopically; that is, adjacent points in the retinal image usually project to adjacent points in striate cortex

皮层是一层神经元，厚1.7毫米，面积1300平方厘米，构成了大脑两个大脑半球的最外层。•每平方毫米的新皮层包含10万个神经元，整个新皮层总共包含大约250亿个神经元，每个神经元通过4000个突触接收输入。•视觉皮层的整体结构是按视网膜局部组织的;即视网膜图像中的相邻点通常投射到纹状皮层中的相邻点

Visual Pathway –Optic Radiations

Simpler Model of Visual Input to Brain: Brain Computing

Spike travels at conduction velocities from 1 to 120 meters (3 to 380 feet) per second . If an insulating myelin sheath (signal booster) is wrapped around the axon then the action potential propagates by “jumping” between gaps in the myelin sheath, otherwise the action potential decays exponentially.

Artificial neuron

weighted input, activation function, output

MCP (McCulloch and Pitts) Neuron – Weights Are Adjusted But Not Learnt (Q3)

learn to calculate

g(z)

`a*w1 + b*w2 + bias*w3` >=0 ->1, else 0;

Turing Test

The 1943 Turing Test was designed to provide a satisfactory operational definition of intelligence.

The test is conducted with two people and a machine

One person plays the role of an interrogator/tester and is in a separate room from the machine and the other person

The interrogator C cannot see the machine and person, he only knows the person and machine as A and B

The aim of the machine is to fool the interrogator into thinking that it is a person.

The interrogator’s task: to find out which candidate is the machine or human, only by asking them questions

If the machine can **fool** the interrogator **30%** of the time, the machine is considered intelligent

If the Turing Test was passed, Turing would conclude that the machine was intelligent.

Suggested as a way of saying when we could consider machines to be intelligent, or at least act intelligently

A satisfactory operational definition of intelligence

Chinese Room

The outsiders act as programmers, the people in the house as computers, and the manuals as computer programs.

L7 Perceptron and Hebb’s Law

Brain-Inspired Intelligence

Perceptrons 感知器

•Single-layer feed forward neural network (perceptron network) 前反馈

–Output units all operate separately: no shared weights

A network with all the inputs connected directly to the outputs

Since each output unit is independent of the others, we can limit our study to single output perceptrons.

Traditional Perceptron

Perceptron Weights are Learned Y= F (W*X)

• Works perfectly if data is linearly separable. If not, it will not converge.

• Idea: Start with a random hyperplane and adjust it using your training data.

• Iterative method.

perception algorithm

```
Input: A set of examples, (x1,y1), ...,(xn,yn)
Output: A perception defined by (w0,w1,...,wd)

Begin:
Initialize the weight wj to 0 for j in range(d)
while(not converge):
    for i in range(n):
        if yi*f(xi)<=0: # an error
            update all wj with wj := wj + yi*xij
```

Perceptron Unit Mimics the Neuron

Inspired by the way **neurons work together in the brain**, the perceptron is a **single-layer neural network** – an algorithm that classifies input into **two** possible categories. The neural network makes a prediction – say, right or left; or dog or cat – and if it’s wrong, tweaks itself to make a more informed prediction next time. It becomes more accurate over thousands or millions of iterations.

Perceptron 6 Components 1 - Input

All the feature becomes the input for a perceptron. We denote the input of a perceptron by [x1, x2, x3, ..,xn], here x represent the feature value and n represent the total number of features. We also have special kind of input called the BIAS

Perceptron 6 Components 2 - Bias

A bias neuron allows a classifier to shift the decision boundary left or right. In an algebraic term, the bias neuron allows a classifier to **translate its decision boundary**. To translation is to "move every point a constant distance in a specified direction". BIAS helps to training the model faster and with better quality.

Perceptron 6 Components 3 - Weight

The weights offer an preliminary value in the very beginning of algorithm learning. With the occurrence of every training inaccuracy, the weights values are updated. These are mainly signified as w1, w2, w3, w4 and so on.

Perceptron 6 Components 4 – Weight Summation

Weighted Summation is the sum of value that we get after the multiplication of each weight [wn] associated the each feature value[xn]. We represent the weighted Summation by ∑wixi for all i -> [1 to n]

Perceptron 6 Components 5 – Transfer Function

Transfer/Step/Activation Function:- the role of activation functions is to make neural networks linear/non-linear. For Perceptron linearly classification of example, it typically uses Heaviside step or Sign function to make the perceptron as linear as possible

Perceptron 6 Components 6 – Output

Output:- The weighted Summation is passed to the step/activation function and whatever value we get after computation is our predicted output. (different classes, -1/1, 0/1, face/none-face, disease/non-disease, etc..)

PLR Foundation - Hebb's Law

When an axon of Neuron A is **near enough to excite** a Neuron B and r**epeatedly or persistently** takes part in firing it, some **growth process or metabolic change** takes place in one or both Neurons such that A's efficiency, as one of the Neurons firing B, is increased

Perceptron Learning Rule 1: PLR (Q4)

1.Randomly choose the weights in the range 0 and 1.

2.Training examples are presented to perceptron one by one from the beginning, and its output is observed for each training example.

3,If the output is correct then the next training example is presented to perceptron.

4.If the output is incorrect then the weights are modified as per the following Perceptron Learning Rule (PLR).

New Wi = Wi + (η * Xi * E).

Change in Weight i = Learning Rate × Current Value of Input i × E (Expected Output, Current Output).

E means error

5.A simple form of E = (Expected Output - Current Output) or SIGN (Expected Output - Current Output).

6.In PLR, output is 1/0 (or -1), and the transfer is **Threshold Step Function**

Rule 2: Perceptron Converge Theorem

The Perceptron convergence theorem states that for any data set which is l**inearly separable** the Perceptron learning rule is guaranteed to find a solution in a finite number of steps

In other words, the Perceptron learning rule is guaranteed to converge to a weight vector that correctly classifies the examples provided the training examples are linearly separable

A function is said to be linearly separable when its outputs can be **discriminated by a function which is a linear combination of features**, that is we can discriminate its outputs by **a line or a hyperplane**.

Traditional Perceptron Decision Surface

A threshold perceptron **returns 1 iff** the **weighted sum** of its inputs (including the bias) is **positive**

Use terminal in VS code to run the after leading project

install node.js and yarn first

then type `yarn -v` to check the installation

```
$ npm install --force
$ npm install craco
$ npm start
```

L8 Mid-Term Review and AI Platform Introduction

Python

Machine learning tools

scikit learn

• Supervised learning: Training data include both inputs and outputs

• Data collection: Start with training data D from which experience is learned.

• Data representation: Encode D to be the input to the learning system.

• Modelling: Choose hypothesis space $\mathcal{H}$ --- a set of possible models for D.

• Learning: Find the best hypothesis $h \in \mathcal{H}$ according to some objective.

• Model selection: Select the best model according to some criteria.

• Two categories: • Classification • Regressio

Train-Test Split

• To evaluate the generalization of the model, we shouldn't use the whole dataset to train the model. We should train the model on a part of the dataset, and test it on the remaining part.

• Use the function train_test_split, we can achieve this easily

聚类与分类的不同在于，聚类所要求划分的类是未知的。 聚类是将数据分类到不同的类或者簇这样的一个过程，所以同一个簇中的对象有很大的相 似性，而不同簇间的对象有很大的相异性

混淆矩阵也称误差矩阵，是表示精度评价的一种标准格式，用n行n列的矩阵形式来表示。

Cross-validation

Deep learning tools

build a model

train: loop: load data, model and get loss, from loss backward to upgrade model

predict: load data, model and predict

1.Initialize 2.Training 3.Validating 4.Testing

L9 ADALINE and BP

Linear Regression

Linear Regression is a statistical procedure that determines the **equation for the straight line** that best fits a specific set of data.

ADALINE (Adaptive Linear Neuron) 自适应线性神经元

ADALINE is an early single-layer artificial neural network based on Least Mean Squares (LMS) algorithms. 最小均方算法

In the **perceptron**, we use the **predicted class labels to update** the weights, and in **ADALINE**, we use **output to update**, it tells us by "how much" we were right or wrong

Adaline algorithm is **identical to linear regression** except for a **threshold function** that **converts the continuous output into a categorical class label**

Widrow Hoff Learning Algorithm (Delta Rule)

$$\Delta W = \alpha x(t-y)$$
$$LMS = (t-O)^2$$
$$w_{t+1} = w_t - \alpha\frac{\partial L}{\partial w_t} = w_t - \alpha(t_t - O_t)x_t$$

α is learning rate, x is input values, and y is output, t is target value

L10 BP and SVM

3 Key Network Components of Neural Network

• **Network Architecture** : single layer feed-forward, multi-layer feed-forward, recurrent network

• **Transfer Function**

Activation (Transfer) Function Extension

sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

tanh

$$\frac{e^x - e^{-x}}{e^x + e^{-x}}$$

(-1,1)

ReLU

max(0,x)

• **Learning Rule**

Back Propagation

Hidden Layers for NN

a NN with 1 hidden layer is a universal function approximator

3 or more hidden layers are hard to train

BP Algorithm with Sigmoid Transfer Function

1. set initial weight and bias a random small number
2. select a sample x from the N input samples and corresponding output y
3. calculate all the output in all layer
4. calculate the output error
5. update the weight for both output and hidden layers
6. t++, repeat 2-5 until the error rate for the N samples reaches the set target or stops decreasing.

Notation 1: BP N Training Samples, p1 Input x ($p1$ features)

$x_i = [x_{i1}, x_{i2}, \ldots\ldots, \ x_{ip1},]^T$

$i = 1, 2, \ldots\ldots, N$

Notation 2: Structure for BP : m Layers

$Y = [y_1^m, y_2^m, \ldots\ldots, y_{p_m}^m]^T$

Notation 3: Weight W_{jk}

W_{jk}^{l} is the weight from the kth neuron in the (l-1)th layer to the jth neuron in the lth layer

Notation 4: u of Layer k Node i in BP

Kth layer, we have P_K inputs, u is the SUM before transfer, f is the activate function

$$u_i = \sum_{j=0}^{p_k - 1} w_{ij}^{k} y_j^{k-1}$$
$$y_0^{k-1} = \theta_i, w_{i0}^{k} = -1$$
$$y_i^{k} = f(u_i^{k})$$

Notation 5: BP error function J and node function u, y

$$J = \frac{1}{2} \sum_{j=1}^{p_m} (y_j^{m} - y_j)^2$$
$$u_i^{k} = \sum_{j} w_{ij}^{k} y_j^{k-1}$$

BP Algorithm Code Flowchart

- Initialize weights (typically random)
- Keep doing epochs
  - For each example in training set do
    - forward pass to compute
      - O = neural-net-output(network, example)
      - miss = (T-O) at each output unit
    - backward pass to calculate deltas (d) to weights
    - update all weights
  - end
- until tuning set error stops improving

BP and Perceptron Weight Updating

1) The none-linear activation of the BP hidden unit is used.

2) The BP rule contains a term for the gradient of the activation function and use **gradient descent** to minimize the error

3) BP can not use Perceptron Learning Rule as no teacher values are possible for hidden units

Gradient Descent Vs Stochastic Gradient Descent

batch needs fewer iterations

Batch Gradient Descent, **对于最优化问题，凸问题**，也肯定可以达到一个**全局最优。**因而理论上来说一次更新的幅度是比较大的。如果样本不多的情况下，当然是这样收敛的速度会更快。但是很多时候，样本很多，更新一次要很久，这样的方法就不合适。

随机梯度下降, 每次更新一个样本，最终的**结果**往往是在**全局最优解附近**

Support Vector Machine

supervised learning models with associated learning algorithms

based on the **statistical learning framework or VC theory**

separate categories are divided by a clear **gap that is as wide as possible**

2 Class Classification: f(x,w,b) = sign(w. x + b)

Margin of a linear classifier is the width that the boundary could be increased by before hitting a data point.

Support Vectors are those data points that the margin pushes up against

The maximum margin linear classifier is the Linear SVM (LSVM)

L11 SVM and Deep Learning

Distance of Point to Line

$$d(x) = \frac{|w^T x + b|}{\sqrt{||w||^2}}$$

Margin m

he decision boundary should be as far away from the data of both classes as possible We should maximize the margin m: smallest distance from observations to hyperplane

Solve SVM by Decision Boundary (Max Margin)

• Let {x1, ..., xn} be our data set and let yi Î {1,-1} be the class label of xi

• The decision boundary should classify all points correctly

• To see this: when y=-1, we wish (wx+b)<1, when y=1, we wish (wx+b)>1. For support vectors, we wish y(wx+b)=1.

• The decision boundary can be found by solving the following constrained optimization problem

Hinge Loss 铰链损失函数

For an intended output t = ±1 and a classifier score y, the hinge loss of the prediction y is defined as

ℓ(y)=max(0,1-t·y)

Note that y should be the "raw" output of the classifier's decision function, not the predicted class label

Kernel in SVM

computing the dot product in some very high dimensional feature space

deal with non-linear data.

SVM Unique Solution

Maximize Margin = Minimize Hinge Loss (y>1, Loss =0)

CNN

feature learning: convolution, activation, pooling

classification: flatten, fully connected, activation

Correlation and Convolution

相关与卷积

当kernel is symmetric, 两种运算一致

卷积是将kernel做关于中心点的对称后的相关

Convolution layer

Characteristics: Ø Local connectivity Ø Parameter sharing

Output Size Computing without Padding

$o = (N - F + 2 * P)/s + 1$

o is the output size, N is the input size(width/height) F is the filter parameter (height/length), s is the stride, p is the padding size

depth of the output = number of filters

Pooling Layer

local or global pooling layers to streamline the underlying computation. (simplification)

reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer

Flatten

Convolution卷积层之后是无法直接连接Dense全连接层的，需要把Convolution层的数据压平（Flatten），然后就可以直接加Dense层了。把 (height,width,channel)的数据压缩成长度为 height × width × channel 的一维数组，然后再与 FC层连接

Fully Connected Layer (FC Layer)

The flattened matrix goes through a fully connected layer to classify the images

given a weight matrix W, output = WI

I is the input matrix

Number of Parameters of a Fully Connected (FC) Layer connected to a Conv Layer

$P = F_{-1} * F + B, B = F$

P is the number of parameters, F_{-1} is the number of neurons in the previous layer(input size), F is the number of neurons in FC layer (output size), B is number of biases in FC layer

LeNet (1998) – The Origin of Convolutional Neural Network CNN

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

VGG Network - Visual Geometry Group, University of Oxford

GoogleNet Inception Module

ResNet

A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or short-cuts to jump over some layers

RNN and LSTM

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs.

Long Short-Term Memory 遗忘门 (forget gate)、输入门 (input gate)、输出门 (output gate)

cell state = forget rate × precious cell state + input × cell memory(cell input activation)

GAN Unsupervised Network

Generative Adversarial Networks (GAN) is one of the most promising recent developments in Deep Learning. GAN, introduced by Ian Goodfellow in 2014, attacks the problem of **unsupervised learning** by training two deep networks, called **Generator and Discriminator**, that **compete and Cooperate with each other**.

Diffusion Model

L12 Deep Learning

ChatGPT and Proximal Policy Optimization(PPO)

Unsupervised Learning

Open Topic: Deep Learning and Machine Learning

Deep Learning Neural Networks replaced handcrafted features with handcrafted architectures

Prior knowledge is not obsolete: it is merely incorporated at a higher level of abstraction.

Active learning: It is about how much you think and learn

Collective study: Let us study together

L13 Project Presentation

nothing new

L14 Final Review

nothing new