

深度学习与中文自然语言处理

Deep Learning for NLP based on TensorFlow



全球人工智能学院

国内首家专注于AI技术职业化教育平台

机器学习原理

机器学习概论

- 1. 什么是机器学习
- 2. 机器学习与传统方法的区别
- 3. 机器学习有什么类型
- 4. 欠拟合与过拟合

- 什么是机器学习
- 问题引导： 图像分类问题 与 自然语言识别问题
- 传统的非机器学习方法： Rule Based
- 图像分类： 点—线—面—连接点—切割点—颜色变化
 - if else:
 - if else:
 - if else:
 -

- 自然语言识别：
- 语言学家的专家知识：
- 字—词—句法—语法—各种词汇出现的频率
- if else
- if else
- if else
- ...

Question

- 基于规则的有什么缺点：
- ☐ A. 规则复杂， 规则直接可能互相冲突
- ☐ B. 不能及时适应新的问题
- ☐ C. 需要反复修改
- ☐ D. 向别人解释困难

- 人类的经验具有不可靠性！
- (世界并不只有一种解释方法)

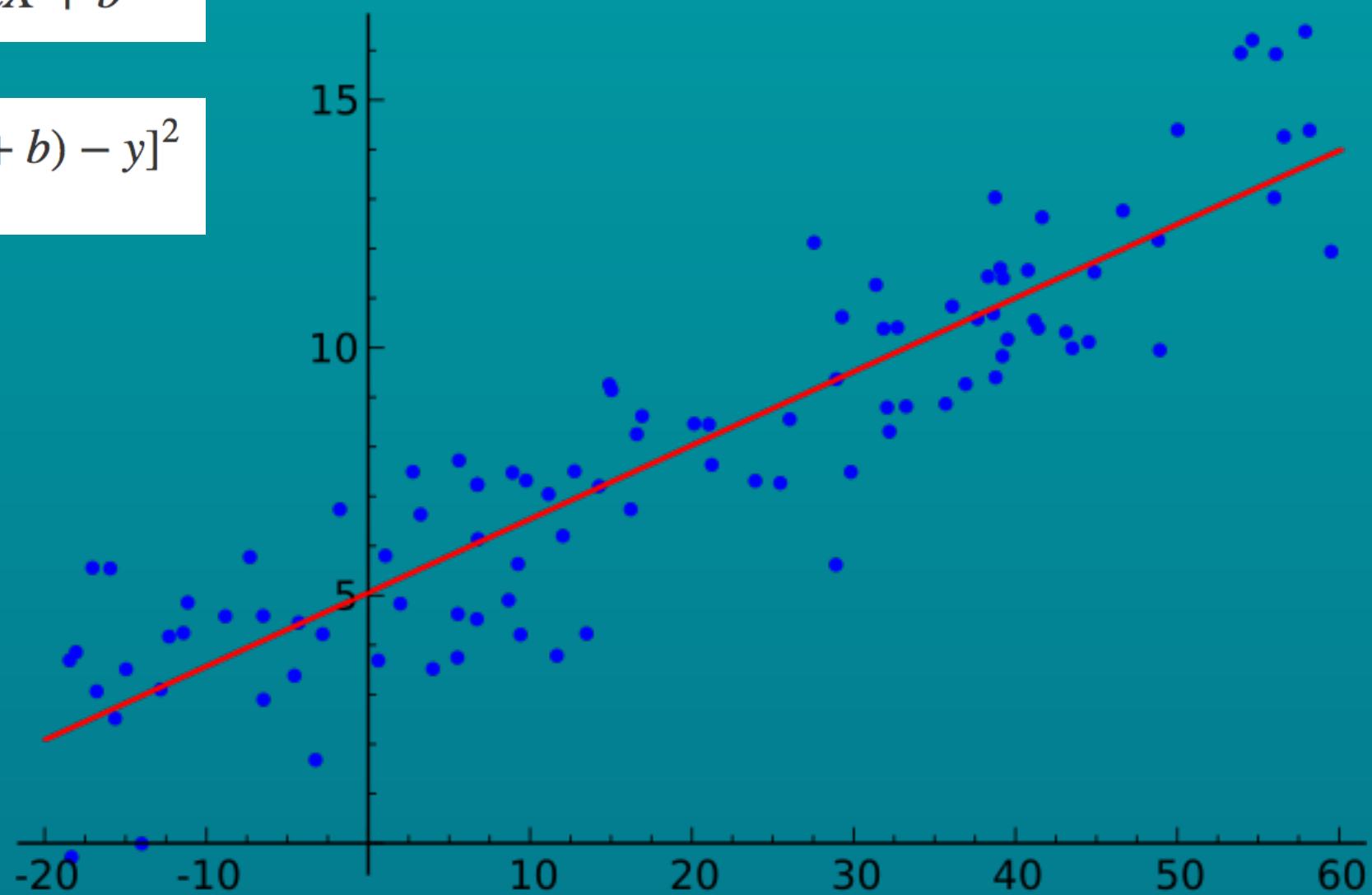
解决办法

- 1. 需要一种能够让程序自动调整的方法
- BUT HOW?

一个例子

$$\hat{Y} = aX + b$$

$$\sum_{x \in X} [(ax + b) - y]^2$$



我们发现了什么？

- 1. 利用计算机反复迭代自动调节； (Train Epoch)
- 2. 给计算机设定目标; (Loss Function)
- 3. 设定更新策略； (Optimizer)
- 4. 使学习出来的参数比较“均匀”； (Regularization)

机器学习工作者的目标

- 1. 建立一个程序(模型, Model), 使得Loss函数能够降低至0;
- 2. 降低至0之后, 能够在其他未曾见过的数据集上取得同样小的Loss
- 3. 能够缩短训练时间;
- 4. 如何使用小样本进行训练;
- 5. 以及其他 (表示学习, active learning)

机器学习的类型

- 1. Classification
- 2. Regression
- 3. Policy & Optimization (Reinforcement Learning)
- 4. Cluster (K-means)
- 5. Generative Model (GAN)

- 机器学习有什么类型
- 1. Classification
- 2. Regression
- 3. Policy & Optimization
- 4. Cluster

欠拟合与过拟合

- 欠拟合： 训练Loss函数不下降
- 过拟合： 训练时候Loss函数下降但是在测试集上效果差

Assignment

- 调研欠拟合和过拟合的原因并查找解决办法
- （机器学习面试高频题目！）
- Next：神经网络的基本原理和变体

深度学习与中文自然语言处理

Deep Learning for NLP based on TensorFlow



全球人工智能学院

国内首家专注于AI技术职业化教育平台

神经网络的基本原理

神经网络的基本原理

- 输入： X
 - X : $\langle x_1, x_2, \dots, x_N \rangle$
- 输出： Y
 - Y : 0.3, 1.4 等数值
 - Y : $[0, 0, 1], [1, 0, 0], [0, 1, 0]$ 等one-hot类型

如何实现？

- 假设X和Y之间满足某种函数关系：
 - $Y = f(X)$
- Y如何表示？

$$f(x) = ax + b$$

$$f(x) = \text{sigmoid}(ax + b)$$

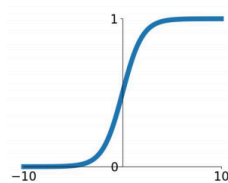
$$f(x) = ax^2 + bx + c$$

$$f(x) = \sin(x) + \cos(x)$$

- 非线性变换: $f(x) = NoLinear_1(a_1 * NoLinear_0(a_0x + b_0) + b_1)$
- 非线性变换能力的叠加: (层数 Layer)
- a_1, a_2 : Weights 权重
- 常用的非线性函数: (activation Function 激活函数)

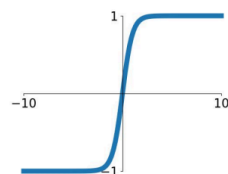
Sigmoid

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



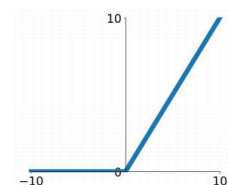
tanh

$$\tanh(x)$$



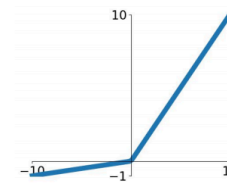
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

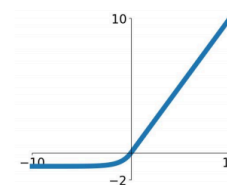


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

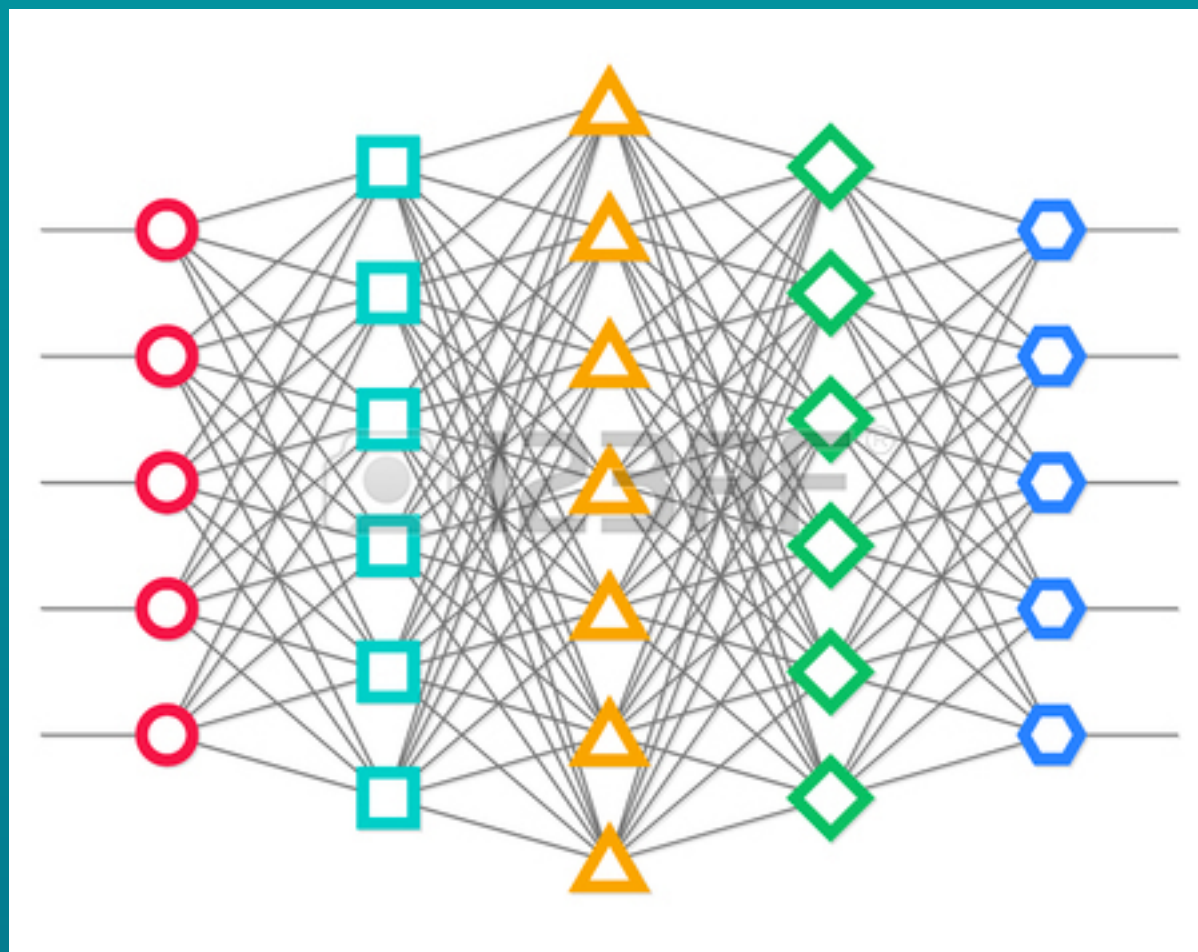
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



图像化表示

- 输入: $X = [x_0, x_1, x_2, \dots, x_n]$



AlexNet

Convolutional network (AlexNet)

input image

weights

loss

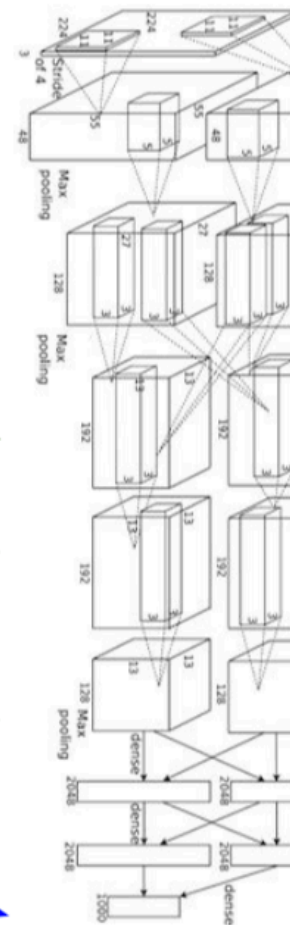


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Assignments

- 1. 复习高等数学“梯度”的概念
- 2. 复习高等数学关于偏导的概念

Next 神经网络如何实现自动优化

深度学习与中文自然语言处理

Deep Learning for NLP based on TensorFlow



全球人工智能学院

国内首家专注于AI技术职业化教育平台

神经网络的基本原理(二)

神经网络的基本原理(二)

- 1. Loss函数
- 2. Backpropagation
- 3. Softmax
- 4. Cross Entropy (交叉熵)

Loss函数

- 1. Regression问题: Loss 定义为:

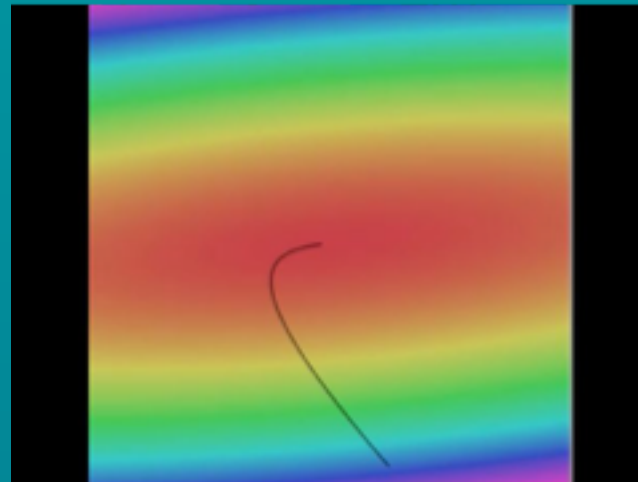
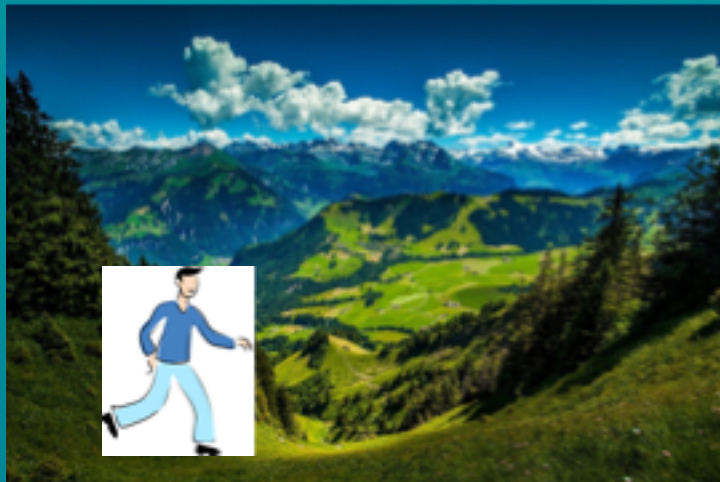
$$Loss(f(\mathbf{x}); \mathbf{y}) = \sum_{i=1}^M (y_i - f(x_i))^2$$

$$f(x) = NoLinear_1(a_1 * NoLinear_0(a_0x + b_0) + b_1)$$

- 2. 问题定义为: 找到一组 $a_0, a_1, b_0, b_1, \dots, a_N, b_N$ 使得Loss最小
 - But How?*

Backpropagation

- 梯度是什么？



$\nabla\varphi$ 或 $\text{grad } \varphi$

其中 ∇ (nabla) 表示向量微分算子。

$\nabla\varphi$ 在三维直角坐标中表示为

$$\nabla\varphi = \left(\frac{\partial\varphi}{\partial x}, \frac{\partial\varphi}{\partial y}, \frac{\partial\varphi}{\partial z} \right)$$

```
while True:
```

```
    weight_grad = evaluate_gradient(loss_fun, data, weights)
```

```
    weights += - step_size * weight_grad
```

Backpropagation

Backpropagation: a simple example

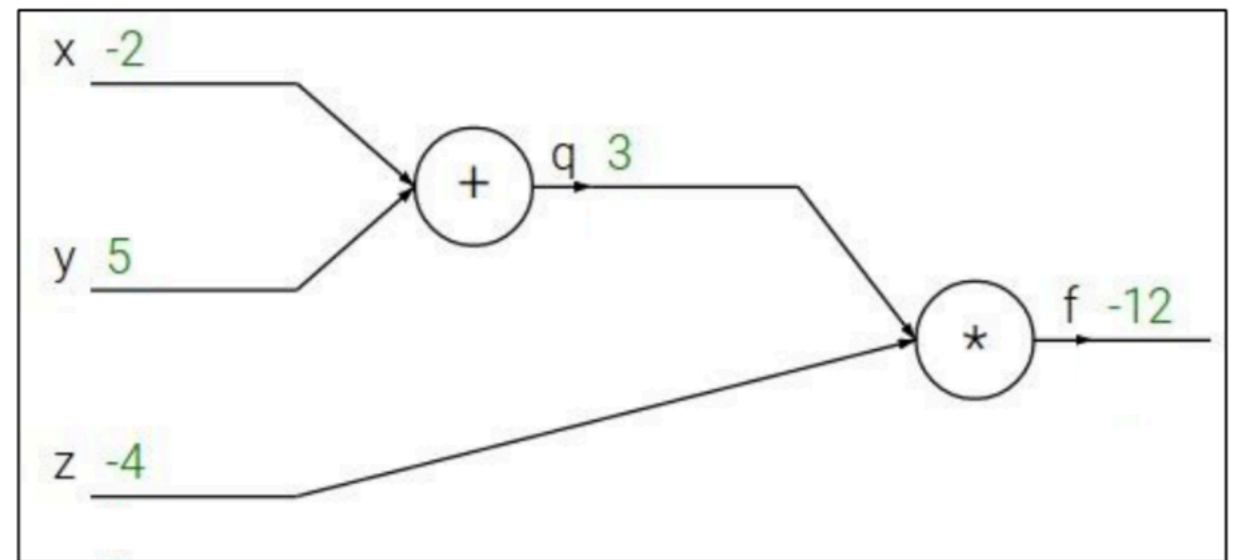
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2, y = 5, z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

Loss函数

- 1. 对于Regression 回归预测问题:

- SSE

$$Loss(f(\mathbf{x}); \mathbf{y}) = \sum_{i=1}^M (y_i - f(x_i))^2$$

- 2. 对于分类问题:
 - softmax + cross entropy

Softmax

- 需要预测的值为一个类型： [0, 0, 0, 1]
- 经过计算， 模型为每一个标签打分： [1.3, -1.3, 0.2, 2.5]

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \dots, K.$$

```
y_hat = [1.3, -1.3, 0.2, 2.5]
y_hat = np.exp(np.array(y_hat)) / sum(np.exp(np.array(y_hat)))
y_hat

array([ 0.21153896,  0.01571176,  0.0704152 ,  0.70233408])
```

将每个预测值变为概率的形式

Softmax

- 获得了每个类型预测的概率
- 我们的目标： 正确的标签的值远远大于其他， 最好为1.0
- 如何进行： 交叉熵

Cross Entropy

- 熵： 物体的混乱程度
- 信息熵：

$$H(X) = \sum_{i=1}^n P(x_i) I(x_i) = - \sum_{i=1}^n P(x_i) \log_b P(x_i),$$

$$- \sum_{c \in C} y_c * \log(\hat{y}_c)$$

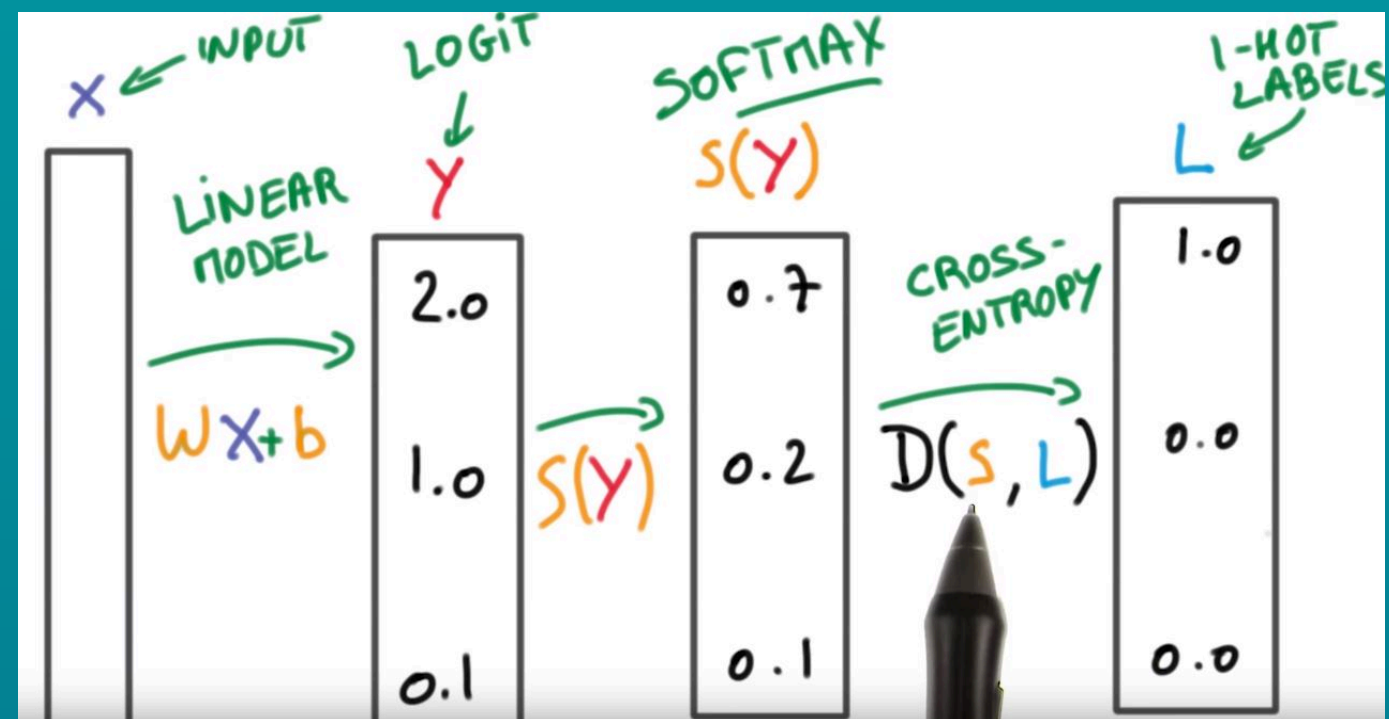


Image source: Google Tensorflow Course

Loss 函数之后?

- 利用backpropagation 进行训练
- 计算获得的梯度为n, 则相应的W更新为
 - $W += -1 * \text{learning_rate} * W$

Loss + 正则

- 为了使得模型中的参数大小比较均衡，真正的 Loss 会加上所有参数的和：

$$Loss = Loss + \lambda \sum_{p \in Parameters} W_p^2$$

如何训练

- 1. 从数据中选取部分数据， 利用模型进行拟合；
- 2. 计算出梯度， 进行反向传播；
- 3. 基于反向传播进行参数的更新；
- 4. 直到模型稳定

神经网络区别于其他机器学习模型的特点

- 1. 模型类别灵活；
- 2. 抽象能力强大；
- 3. 所需的数据量大；
- 4. 泛化能力强；

Assignment

- 1. 为什么数据在输入到模型之前要进行normalization 和 scaling?
- normaliaztion 使得标准差比较小的值（例如1），平均数为0；
- scaling 使得所有的值都在一定的范围内
- **调研这是为什么？**
- 2. 安装tensorflow 版本 ≥ 1.2
- Next: Tensorflow 的基本结构

深度学习与中文自然语言处理

Deep Learning for NLP based on TensorFlow



全球人工智能学院

国内首家专注于AI技术职业化教育平台

用Tensorflow实现神经网络

用Tensorflow实现神经网络



- 1. 神经网络的基本构成；
- 2. Tensorflow框架；

神经网络的构成

- 1. 训练数据
- 2. 参数
- 3. Operator 计算符
- 4. Loss 函数
- 5. 优化方法 (SGD, Adam等)

神经网络框架

- 1. 常用的方法框架已经定义好;
- 2. 可以自定义网络结构;
- 3. Tensorflow, Caffe, PyTorch
 - 自动计算导数\梯度
 - 自动根据梯度进行参数的迭代优化

TensorFlow—Tensor

tensor

't'
'e'
'n'
's'
'o'
'r'

tensor of dimensions [6]
(vector of dimension 6)

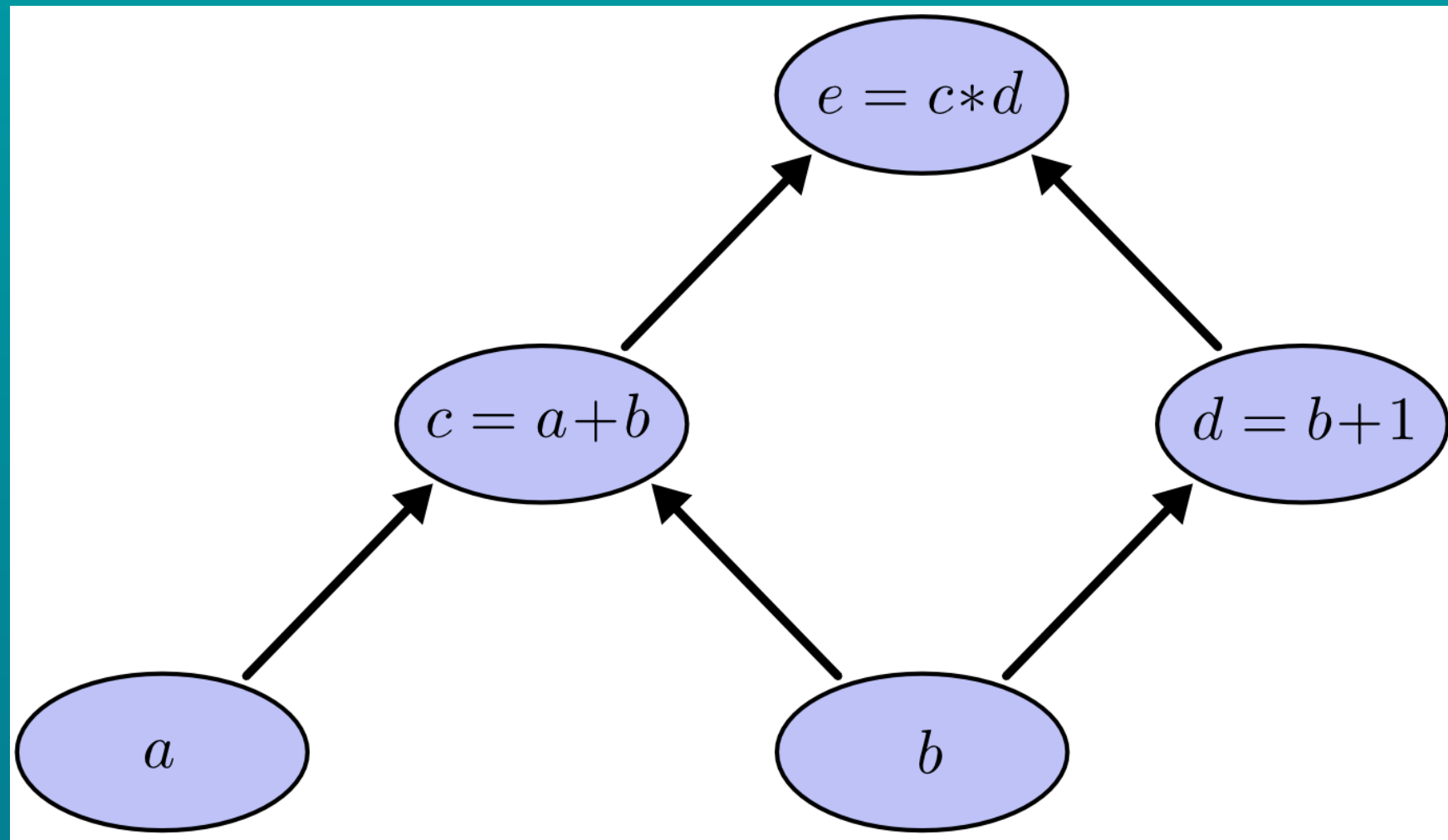
3	1	4	1
5	9	2	6
5	3	5	8
9	7	9	3
2	3	8	4
6	2	6	4

tensor of dimensions [6,4]
(matrix 6 by 4)

2	7	8	8	1	8
2	8	4	5	9	0
2	3	5	3	6	0
7	4	7	1	3	5

tensor of dimensions [4,4,2]

Tensorflow-GraphFlow



Session 和 Graph

- 所以Tensorflow中tensor的值， 需要在session中进行计算！
 - 实例： session, ipython-notebook
- Graph 定义了一个计算图， 一整个模型可以有好几个sub-graph构成， 不同的graph用来计算不同的模型图

TensorFlow的元素

- 1. 数据： 两种一种是利用feed_dict进行， 另一种是利用Data.pipeline 进行.
- 2. Variable： 变量， 能够在训练中改变；
- 3. placeholder： 位置的量， 例如训练数据， 需要每次导入；
- 4. Operator: 各种数学操作
- 5. Optimizer： 进行自动优化的函数

深度学习与中文自然语言处理

Deep Learning for NLP based on TensorFlow



全球人工智能学院

国内首家专注于AI技术职业化教育平台

利用神经网络+word2vec进行文本分类

利用神经网络+word2vec进行文本分类

- 1. 基本原理
- 2. 示例程序讲解

基本原理

- 1. 将词汇变成one-hot
- 2. 利用one-hot选取单词的词向量
- 3. 将单词的词向量拼接变成句子向量
- 4. 使用模型进行分类

实例程序讲解

- 使用Tensorflow对豆瓣电影的评论进行自动分类

Assignment

- 1. 查明以下单词的意思： fully connected layer, regularization, logits;
- 2. 思考： 进行分类问题的时候， 最后一层为什么要用全连接；
- 3. 用tensorflow实现文本分类的任务， 数据集从kaggle里边获得；