Alex Meng's Notes

Alex Meng

2025-06-06

Table of contents

Preface

If you are reading this, you may be interesed in seeing what is "Alex's Notes".

These notes are just things that I am documenting, that I wish could become a useful resource for my future students, either when I TA or become a professor.

Here's how to learn anything:

1. Write it out! (Document, Code along)
2. EXPERIMENT and explore
3. Visualize things you don't understand
4. Ask Questions
5. Answer Exercise and Problems (stretch your knowledge)
6. Share with like-minded individuals

Deep Learning Systems

In this set of notes, we will create a minimal version of PyTorch / Tensorflow from scratch, of what I call "PyStickOnFire".

Chapter 1: Machine Learning Refresher

The (Supervised) Machine learning idea: we take a bunch of labeled data, feed them to a machine learning algorithm, and it outputs a "program" that solves the task.

B[Machine Learning Algorithm] B –> C(Model h)

```
We will focus on what the machine learning algorithm box contains. In general, it consists o

1. The hypothesis class (the structure of $h$ in terms of a set of parameters)
2. The loss function (specifies how good a given hypothesis is)
3. An optimization method (the way to minimize the loss function)

All alogrithms in machine learning fit in this structure. Let's look at **softmax regression

### Multi-class Classification (Softmax Regression)

Consider a *k-class classification setting*, where we have


* training data:
  $$x^{(i)} \in \mathbb{R}^{n}, y^{(i)} \in \{1, \dots, k \} \text{ for } i = 1, \dots, m$$
  * $n =$ dimensionality of input data
  * $k =$ number of different classes / labels
  * $m =$ number of data points in the training data

where the training data are vectors that looks like
$$X = \{ \begin{bmatrix} x^{(1)}_1 \\ x^{(1)}_2 \\ \vdots \\ x^{(1)}_n \end{bmatrix} , \dots
and the labels are just a set of scalars of size $k$.
```

```
### 1st Element: The Hypothesis Function

The hypothesis function is a mapping from one input to one output. (Duhh... just like every c
$$h: \mathbb{R^n} \rightarrow \mathbb{R^k}$$
$$h(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_k(x)\end{bmatrix}$$

So what really is $h_i(x)$? It is the hypothesis, the "belief", the probability of how likely

A **linear hypothesis function** uses matrix multiplication, or some other linear way, for th

$$h_\theta(x) = \theta^T x$$

for parameters $\theta \in \mathbb{R^{n \times k}}$ ($n$ rows and $k$ columns, so transpose h

Notice how so far we only have one input and one output, $h$ is only working on one instance

$$X \in \mathbb{R^{m \times n}} = \begin{bmatrix} -x^{(1)^{T}}- \\ -x^{(2)^{T}}- \\ \vdots \\

$$y \in \{ 1, \dots, k \}^m = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)}  \end{b

$$h_\theta(X) = \begin{bmatrix} -h_\theta(x^{(1)})^{T}- \\ -h_\theta(x^{(2)})^{T}- \\ \vdots

  * $n =$ dimensionality of input data
  * $k =$ number of different classes / labels
  * $m =$ number of data points in the training data

Each row is for a data point, the first example is in first row (originally a column vector,
```

```
### 2nd Element: The Loss Function

How are we going to evaluate the quality of our predictions?

**Classification Error**

$$l_{err}(h(x),y) = \begin{cases} 0, & \text{if } argmax_i h_i(x) = y \\ 1, & \text{otherwis

The error is not differentiable, so it is not good for optimization.

**A better choice: Cross-Entropy Loss or Softmax**

The idea is that we want to map our outputs into being actual probabilities

$$h_i(x) \rightarrow prob[label == i]$$

Probability has to be positive and sum to 1. In order to ensure $h_i(x)$ is positive, we can

$$prob[label == i] = normalize(\exp(h(x)))= \frac{\exp(h_i(x))}{\sum_{j = 1}^{k} \exp(h_j(x)

This is called the **softmax operation**, a mapping between scalar values and a probability d

So now we have a probability, We need some way of quantifying whether the vector of probabili

$$l_{cross-entropy} (h(x), y) = - prob[label = y]$$

Because minimizing probabilities is not numerically good: Probabilities are bounded between 0

$$l_{ce} (h(x), y) = - log(prob[label = y]) = -h_y(x) + log \sum_{j=1}^{k} exp(h_j(x))$$

This is commonly known as the **negative log loss** or **cross-entropy loss**. This is also a
```

```
### 3rd Element: Optimization

How do we find good values for $\theta$?

This element we will spend the most time to cover because we not only what to know what the c

*The following problem is the problem that almost all machine algorithms are solving.* Here i

$$
\min_{\theta} \frac{1}{m} \sum_{i=1}^{m} l(h_\theta(x^{(i)}), y^{(i)})
$$

We are searching over all possible values of $\theta$, the one that minimizes the *average* l

$$
\min_{\theta} f(\theta) = \min_{\theta} \frac{1}{m} \sum_{i=1}^{m} l_{ce}(\theta^T x^{(i)}, y
$$

*Now we know the "what", but, how do we find that?* How do we solve $\min_{\theta} f(\theta)

#### The Gradient

For our $f(\theta)$ (the function that we are trying to minimize), remember, this function ta
$$f: \mathbb{R^{n \times k}} \rightarrow \mathbb{R}$$

$$f(\theta) \in \mathbb{R}$$

Remember that the gradient is a multidimensional derivative that has a direction, which point

Here is the definition of the gradient in our case,

$$\nabla_\theta f(\theta) = \begin{bmatrix}
\frac{\partial f}{\partial \theta_{11}} \frac{\partial f}{\partial \theta_{12}} \dots \frac{\
\frac{\partial f}{\partial \theta_{21}} \frac{\partial f}{\partial \theta_{22}}  \dots \frac
\vdots \\
\frac{\partial f}{\partial \theta_{n1}} \frac{\partial f}{\partial \theta_{n2}}  \dots \frac
\end{bmatrix} \in \mathbb{R^{n \times k}}$$

The derivative of a function is the slope of the function, change in $y$ over change in $x$.
```

```
#### Gradient Descent

If the gradient points in the direction of maximum increase, to minimize a function, we can i

$$\theta = \theta - \alpha \nabla_\theta f(\theta)$$

where $\alpha$ is called the *learning rate* or *step size*. Note that the learning rate must

The choice of the step size $\alpha$ is really really really important. Too small slows down

#### Stochastic Gradient Descent

We split up the dataset into *minibatches*, which are subsets of data of size $B$.

We repeat the process of sampling minibatches and taking steps to update $\theta$.

* Sample: $X \in \mathbb{R^{B \times n}}, y \in \{ 1, \dots, k\}^B$
* Update: $\theta = \theta - \frac{\alpha}{B} \sum_{i=1}^{B} \nabla_\theta l(h_\theta(x^{(i)}

#### Calculating the gradient in practice

In order to calculate the gradient of $f(\theta)$, which is essentially the sum of gradients

$$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} l(h_\theta(x^{(i)}), y^{(i)}) =  \frac{1}{m} \sum_

We have to calculate the gradient $m$ times, which is very expensive. *Can we reduce the numb

As an example, how do we compute the gradient for softmax objective? We can do it by hand, bu

In practice, we just specify the hypothesis function and the loss function, and use **Automat

We can either do it through the "right" way, use matrix differential calculus, jacobians, kr

$$\frac{\partial}{\partial \theta} l_{ce} (\theta^T x, y)= \frac{\partial l_{ce}} (\theta^T x
```

````
`<!-- quarto-file-metadata: eyJyZXNvdXJjZURpciI6ImNvdXJzZXMifQ== -->`{=html}

```{=html}
<!-- quarto-file-metadata: eyJyZXNvdXJjZURpciI6ImNvdXJzZXMiLCJib29rSXRlbVR5cGUiOiJjaGFwdGVyI
````
````

Optics

Q1: One Dimensional Wave Equation

The wave equation is given by

$$\psi(x,t) = \frac{3}{[10(x-vt)^2+1]}$$

Show, using brute force, that this is a solution to the one dimensional differential wave equation.

Great! Let's start with what is a wave.

Def. *A classical traveling wave is a self-sustaining disturbance ψ of a medium, and the disturbance ψ moves through space transporting energy and momentum.*

Everything is waves.

Sound! A type of **longitudinal** wave, where the displacement vector points parallel to the direction of motion.

Guitar string! A type of **transverse** wave, where the displacement vector points perpendicular to the direction of motion.

A wave is not a stream of particles! Because the individual atoms stay in equilibrium, but only the disturbance advances through them. Leonardo da Vinci was one of the first person to realize waves does not transport the medium through which it travels.

Imagine disturbance ψ moves in positive direction x with constant velocity v.

$$\psi = f(x,t)$$

What is $f(x,0)$? it is the shape (aka the **profile**) of ψ at $t=0$. For example, try visualizing $f(x) = e^{-ax^2}$, you'll see that it is a **gaussian function**. Setting $t=0$ is taking a snapshot of the pulse as it travels by.

In order to understand this better, let's ignore t by introducing a coordinate system $S^{'}$ that travels with the pilse at the speed v. As we move with $S^{'}$, the wave looks stationary! So

$$\psi = f\left(x^{'}\right)$$

where $x^{'} = x - vt$, because after time t the same point on ψ moved a distance of vt.

General Form of One Dimensional Wave Function

$$\psi(x,t) = f(x - vt)$$

Jean Le Rond d'Alembert was the one that brought partial differential equations to physics and formulated the differential wave equation.

Quant

Being a quant is knowing how to solve problems with logic, math, and intuition. These Problems come from A practical guide to Quantitative Finance Interviews by Xinfeng Zhou.

1 Problem Simplification

Screwy Pirates

Five pirates looted a chest full of 100 gold coins. Being a bunch of democratic pirates, they agree on the following method to divide the loot:

The most senior pirate will propose a distribution of the coins. All pirates, *including the most senior pirate*, will then vote. If at least 50% of the pirates (3 pirates in this case) accept the proposal, the gold is divided as proposed. If not, the most senior pirate will be fed to shark and the process starts over with the next most senior pirate… The process is repeated until a plan is approved. You can assume that all pirates are perfectly rational: they want to stay alive first and to get as much gold as possible second. Finally, being blood-thirsty pirates, they want to have fewer pirates on the boat if given a choice between otherwise equal outcomes.

How will the gold coins be divided in the end?

Answer

I have no idea what the five pirates will do, lemme consider a simpler case, 1 pirate.

1 pirate. Pirate 1 propose to distribute all 100 gold coins to himself, and accept the proposal.

100 coins to pirate 1

2 pirates. Pirate 2 proposes to get all the gold, 50% good, gets all the gold.

100 coins to pirate 2

3 pirates. From the perspective of pirate 1, pirate 1 gets nothing if pirate 3 pirate 3 gets voted out (back to case 2), so pirate 1 will try to make pirate 3 win, iff pirate 1 gets at least some

benefits. Pirate 3 knows that, so pirate 3 will give 1 coin to pirate 1 and 99 coins to pirate 3, since pirate 1 will think anything is better than nothing.

$$1 \text{ coin to pirate } 1, 99 \text{ coins to pirate } 3$$

4 pirates. If pirate 4 gives to pirate 2, pirate 2 will vote for pirate 4 because if he doesn't, it will be back to 3 pirates case where he doesn't get anything...So

$$1 \text{ coin pirate } 2, 99 \text{ coins pirate } 4$$

5 pirates. Pirate 5 will give coinds to pirate 1 and pirate 3, because doing that will allow them to get some coins, where if he gets voted out, in 4 pirates case they don't get anything...So

$$\boxed{1 \text{ coin for pirate } 1, 1 \text{ coin for pirate } 3, 98 \text{ coins for pirate } 5}$$

We can actually formualte a generalizable law from this using mathematics.