This are my notes and observations from reading the Linear Algebra chapter of the Deep Learning book.

The following notes are presented in order of value to the reader. I start with a discussion of the Moore-Penrose pseudoinverse, followed by a short reflection on "broadcasting".

Moore-Penrose pseudoinverse

The Moore-Penrose pseudoinverse is defined as:

$$A^+ := \lim_{\alpha \searrow 0} \left(A^T A + \alpha I\right)^{-1} A^T$$

You can click on the formula to jump into the book. (It's really awesome that all of it is online and you can deep link to pages.)

This formula is mystifying. What is its origin? And why does it do what the book says it does?

We can get a feeling for its origin by looking at the equation $Ax = b$ that it solves:

$$\begin{aligned} Ax &= b | A^T \cdot \\ A^T Ax &= A^T b | \left(A^T A\right)^{-1} \cdot \\ \left(A^T A\right)^{-1} A^T Ax &= \left(A^T A\right)^{-1} A^T b \\ x &= \left(A^T A\right)^{-1} A^T b \end{aligned}$$

So we start with the equation that we want to solve and find a more complex form for solving for x than the usual left-multiplication with the inverse A^{-1}. This is already useful if $A^T A$ is invertible and A is not square (because matrix inverses only exist for square matrices).

This already looks similar to the pseudo-inverse, except for the $+\alpha I$ term.

Aside: there is also a right pseudo-inverse

We can guess its form by starting with:

$$\begin{aligned} yA &= b && | \cdot A^T \\ yAA^T &= bA^T && | \cdot \left(AA^T\right)^{-1} \\ yAA^T \left(AA^T\right)^{-1} &= bA^T \left(AA^T\right)^{-1} \\ y &= bA^T \left(AA^T\right)^{-1} \end{aligned}$$

From this, we can make an educated guess:

$$^{+}A = A^T \lim_{\alpha \searrow 0} \left(AA^T + \alpha I\right)^{-1}$$

Since we cannot always invert A^TA (respectively AA^T), an obvious question is:

Why can we invert $AA^T + \alpha I$ for positive α?

Let's examine this. For a matrix to be invertible, its kernel has to only contain the zero vector:

$$\ker\left(A^TA + \alpha I\right) = \{0\}$$

To prove that this is the case for $\left(AA^T + \alpha I\right)$, we need to show that:

$$\left(A^TA + \alpha I\right) v = 0 \implies v = 0$$

So starting with the left side, we can rephrase it as follows:

$$\begin{aligned} & \left(A^TA + \alpha I\right) v = 0 \\ \Leftrightarrow \quad & A^TAv + \alpha Iv = 0 \\ \Leftrightarrow \quad & A^TAv = -\alpha v \end{aligned}$$

If $v \neq 0$, this would mean that $-\alpha$ is a negative eigenvalue of A^TA as α is > 0.

Can A^TA have negative eigenvalues?

Let's assume $v \neq 0$ and $-\alpha$ is a negative eigenvalue, that is $A^TAv = -\alpha v$ holds (and $\alpha > 0$). We can left-multiply with v^T and obtain:

$$\begin{aligned} & \Leftrightarrow v^TA^TAv = v^T\left(-\alpha v\right) \\ & \Leftrightarrow \|Av\|_2^2 = -\alpha v^Tv = -\alpha \|v\|_2^2 \\ & \Leftrightarrow -\alpha = \frac{\|Av\|_2^2}{\|v\|_2^2} \geq 0 \end{aligned}$$

(We can divide by $\|v\|_2^2$ because we assume $v \neq 0$.) Now this means, that $-\alpha$ has to be > 0, so it is not a negative eigenvalue and a contradiction to our initial assumption $\alpha > 0$. In fact, we have just shown that A^TA in general can only have non-negative eigenvalues. (We can show the same for AA^T the same way.)

Because A^TA cannot have negative eigenvalues, this means that v cannot be $\neq 0$, so $v = 0$. This is what we wanted to show, and this means that the kernel only contains the zero vector, and $AA^T + \alpha I$ is invertible for $\alpha > 0$.

What if $A^T A$ was invertible?

For a moment, let's consider the case when $A^T A$ is invertible: Is the Moore-Penrose pseudoinverse then equal to $\left(A^T A\right)^{-1} A^T$?

Matrix inversion is continuous in the space of invertible matrices. You might remember the definition of continuity from school. A better definition of continuity is that it means being able to swap function and limit application:

> A function f is continuous on its domain iff for any convergent series $x_n \to x$ for which $f(x)$ is defined
>
> $$\lim_{n\to\infty} f(x_n) = f\left(\lim_{n\to\infty} x_n\right) = f(x).$$

Assuming $A^T A$ is invertible and using the fact that matrix inversion is continuous for invertible matrices, we now see:

$$\begin{aligned}\lim_{\alpha\searrow 0}\left(A^T A + \alpha I\right)^{-1} A^T &= \left(\lim_{\alpha\searrow 0} A^T A + \alpha I\right)^{-1} A^T \\ &= \left(A^T A + 0\, I\right)^{-1} A^T \\ &= \left(A^T A\right)^{-1} A^T\end{aligned}$$

So in this special case, the pseudoinverse is exactly the solution we have come up with ourselves.

Two properties of the pseudoinverse

There are two properties mentioned in text that are interesting but not obvious:

> When A has more columns than rows, then solving a linear equation using the pseudoinverse provides one of the many possible solutions. Specifically, it provides the solution $x = A^+ y$ with minimal Euclidean norm $\|x\|_2$ among all possible solutions. When A has more rows than columns, it is possible for there to be no solution. In this case, using the pseudoinverse gives us the x for which Ax is as close as possible to y in terms of Euclidean norm $\|Ax - y\|_2$.

These properties are not obvious and their deduction is enlightening towards the chosen definition of the pseudoinverse, specifically the use of the limit and the constraint of α to be > 0.

A related optimization problem

To prove the properties, let's look at the following regularized minimum-least-squares problem. This is something that is formulated in more detail in Chapter 4 of the book, but it is quite useful here:

$$\min_x ||Ax - b||_2^2 + \alpha \|x\|_2^2, \alpha > 0$$

Let's solve this optimization problem. First, we set up a cost function:

$$\begin{aligned} c_\alpha(x) &= \|Ax - b\|_2^2 + \alpha \|x\|_2^2 \\ &= (Ax - b)^T (Ax - b) + \alpha x^T x \\ &= x^T A A x - 2b^T A x + b^T b + \alpha x^T x \end{aligned}$$

The first derivative of c_α is:

$$\nabla c_\alpha(x) = 2A^T A x - 2A^T b + 2\alpha x$$

And the second derivative is:

$$H_\alpha(x) = 2A^T A + 2\alpha I$$

H_α is positive definite, that is $v^T H_\alpha(x) v \geq 0$ for all v. Why? We have already seen that $A^T A$ only has non-negative eigenvalues, so it is positive semidefinite by definition and αI is trivially positive definite for $\alpha > 0$. The sum of the two is positive definite again. Thus c is a strictly convex function. This is along to lines of the one-dimensional case: when the second derivative is > 0 everywhere, the function is strictly convex. Convex functions have a global minimum and, for strictly convex functions, this global minimum is unique. So we know there is only exactly one point that minimizes the cost function c_α.

We can determine this global minimum x_α^* by solving $\nabla c_\alpha(x) = 0$:

$$\begin{aligned} & \nabla c_\alpha(x) = 0 \\ \Leftrightarrow \quad & A^T A x - A^T b + ax = 0 \\ \Leftrightarrow \quad & \left(A^T A + \alpha I\right) x = A^T b \\ \Leftrightarrow \quad & x = \left(A^T A + \alpha I\right)^{-1} A^T b \end{aligned}$$

This is exactly the definition of the pseudoinverse without the limit. So we have found:

$$*arg\,min x_\alpha^* =_x ||Ax - b||_2^2 + \alpha \|x\|_2^2 = \left(A^T A + \alpha I\right)^{-1} A^T b$$

with $c_\alpha(x_\alpha^*) \leq c_\alpha(x)$ for all x, and x_α^* denotes the minimum point.

This expression is continuous in α, so we can take the limit $\alpha \searrow 0$. Remember, we need the constraint of $\alpha > 0$ for $A^T A + \alpha I$ to be invertible, so we can only take the limit from above. Taking the limit, we obtain:

$$*arg\,min x^* = \lim_{x \searrow 0} \left(A^T A + \alpha I\right)^{-1} A^T b$$

with $c(x^*) \leq c(x)$ for all x with

$$c(x) = \lim_{\alpha \searrow 0} \|Ax - b\|_2^2 + \alpha \|x\|_2^2 = \|Ax - b\|_2^2 .$$

What do we gain from this? Well for one, we now know that this expression minimizes $||Ax - b||_2^2$. Furthermore, if there is a solution for $Ax = b$, we can easily use a similar approach to see that the solution x^* is smaller under Euclidean norm than any (other) solution $\hat{x}$ for $Ax = b$.

We do this in two steps. First, we observe that

$$A\hat{x} = b \Leftrightarrow A\hat{x} - b = 0 \Leftrightarrow \|A\hat{x} - b\|_2^2 = 0$$

and, because x_α^* minimizes c_α,

$$\begin{aligned} & c_\alpha(x_\alpha^*) \leq c_\alpha(\hat{x}) \\ \Leftrightarrow \quad & c_\alpha(x_\alpha^*) \leq \|A\hat{x} - b\|_2^2 + \|\hat{x}\|_2^2 = 0 + \alpha \|\hat{x}\|_2^2 \\ \Leftrightarrow \quad & c_\alpha(x_\alpha^*) \leq \alpha \|\hat{x}\|_2^2 \end{aligned}$$

This expression is again continuous, so we can take the limit $\alpha \searrow 0$, and we see:

$$\begin{aligned} & c(x^*) \leq 0 \\ \Leftrightarrow & \|Ax^* - b\|_2^2 \leq 0 \end{aligned}$$

Because norms are always non-negative, we have $0 \leq \|Ax^* - b\|_2^2 \leq 0$, so $\|Ax^* - b\|_2^2 = 0$. And we have observed above that this is equivalent to $Ax^* = b$. So if there is at least one exact solution to the problem, we are sure to obtain an exact one, too. To be fair, we could have deduced this in the previous section. However, we can take another limit on the inequality $c_\alpha(x_\alpha^*) \leq \alpha \|\hat{x}\|_2^2$ and obtain a more interesting result. This time, we only take the limit $\alpha \searrow 0$ of x_α^*, but keep c_α fixed:

$$\begin{aligned} & c_\alpha(x_\alpha^*) \leq \alpha \|\hat{x}\|_2^2 \\ \Rightarrow \quad & c_\alpha\left(\lim_{\alpha \searrow 0} x_\alpha^*\right) \leq \alpha \|\hat{x}\|_2^2 \\ \Leftrightarrow \quad & c_\alpha(x^*) \leq \alpha \|\hat{x}\|_2^2 \\ \Leftrightarrow \quad & \|Ax^* - b\|_2^2 + \alpha \|x^*\|_2^2 \leq \alpha \|\hat{x}\|_2^2 \\ \Rightarrow \quad & \alpha \|x^*\|_2^2 \leq \alpha \|\hat{x}\|_2^2 \\ \Leftrightarrow \quad & \|x^*\|_2^2 \leq \|\hat{x}\|_2^2 \end{aligned}$$

Here, we use $\|Ax^* - b\|_2^2 \geq 0$ to be able to drop the term and $\alpha > 0$ to preserve the direction of the inequality, and we obtain the second property about the length of $\|x^*\|$.

Now we have proved both properties that were mentioned in the book. On the one hand, if there are solutions, we will obtain one with minimal norm. On the other hand, if there are no solutions, using the pseudoinverse will provide us with an x that at least minimizes $||Ax - b||_2^2$.

To convince yourself that the expressions above are indeed continuous, we can use the technical argument that we can rewrite $c_\alpha(x)$ into $c(\alpha, x)$ and see that it is a continuous function in two variables instead of going the route of using functional analysis and treating c_α as a convergent series of functions.

Broadcasting notation: oh my...

The broadcasting notation in the Deep Learning book is weird. For one moment, let's ignore its origins from numpy, and let's look at what's happening.

> In the context of deep learning, we also use some less conventional notation. We allow the addition of matrix and a vector, yielding another matrix: $C = A+b$, where $C_{i,j} = A_{i,j}+b_j$. In other words, the vector b is added to each row of the matrix. This shorthand eliminates the need to dene a matrix with b copied into each row before doing the addition. This implicit copying of b to many locations is called broadcasting.

Let's say $A \in \mathbb{R}^{3\times 3}$ and $b \in \mathbb{R}^3$, for example:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Then, with broadcasting:

$$A + b = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}$$

How do we get there? Essentially, we take the vector $b \in \mathbb{R}^3$, interpret it as a row vector, and add it to every row of the matrix, so:

$$A + b = A + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} b^T$$

Wouldn't it make more sense to write this as $A + b^T$? The reason for the unintuitive notation lies in the details of broadcasting: whereas in maths, a vector $\mathbb{R}^3$ is identified as a column matrix $\mathbb{R}^{3\times 1}$, in the context of broadcasting it is treated as a row matrix $\mathbb{R}^{1\times 3}$. This row matrix is then repeated across the 1 dimension to make it into a 3×3 matrix that can be applied to A. However,

for matrix multiplications, b continues to be treated as column matrix. This is really confusing. So what's the reason for this ambiguity?

My best guess is that in ML contexts, data entries (examples) are often treated as row vectors. For example, a design matrix stores a number of examples, each in a separate row. So if you want to change the bias of your examples, you want to add a bias vector as row-vector to each row. Indeed, in chapter 8, broadcasting is used to express batch normalization in eq (8.35):

$$H' = \frac{H - \mu}{\sigma}$$

I don't think this excuses the lack of clarity introduced with this notation, but then again it is too late to change and make everybody use a transposed design matrix... :)

Looking back and ahead

There is much more I could write about, but I think these were the most interesting bits from my notes. We have revisited convex functions, continuity and limits to motivate the curious definition of the Moore-Penrose pesudoinverse. Last but not least, we have looked at the origin of the broadcasting notation and why it might be confusing to someone who is new to ML. Next up: reading Chapter 3 and thinking about probability theory.

Stay tuned,
Andreas

PS: This a gist repost of a post on my blog http://blog.blackhc.net/2017/03/dlb-chapter2/index.html. I wish Medium was supporting LaTeX formulas properly...