Mathematical tools for deep learning

- Manipulating probabilities
- Calculus chain rule
- Gradients
- Gradient chain rule
- Linear algebra intuitions

The purpose of these notes

- These notes covers mathematical tools and intuitions we ought to be familiar with in deep learning.
- Many of these tools are also relevant for work in machine learning and statistical signal processing.

A few words on probability notation

Sometimes, notation in probability can get *sloppy* and it's hard to know exactly what is meant by an expression. We use the following notation in this class.

- A random variable, X, is denoted by a capital letter, and the values it can taken on, x, are denoted by lower case letters.
- Their corresponding vector forms would be $\mathbf{X}$ and $\mathbf{x}$.

The probability that X takes on the value x is a number that we denote as:

$$\Pr(X = x)$$

The inputs to the $\Pr(\cdot)$ function are *events* (like X taking on the value x) and it returns a real-valued number between 0 and 1. This number represents the probability of the event happening.

Thus, $\Pr(X = x)$ is the probability that a random variable X takes on the value x.

A few words on probability notation (cont.)

This notation, though clear, is often cumbersome, and thus shorthands are adopted. One still clear notation is:

$$p_X(x) \triangleq \Pr(X = x)$$

Here, the subscript X denotes the random variable we are evaluating the probability of and x denotes the value it takes on. p_X is typically referred to as the probability mass (density) function when X is a discrete (continuous) random variable.

Still more common is the following notation:

$$p(x) \triangleq \Pr(X = x)$$

This notation *may* be ambiguous, since the random variable is omitted. However, we will operate on the assumption that if the random variable is omitted, we are referring to the corresponding uppercase variable.

A few words on probability notation (cont.)

In this class, we will use the notation:

$$p(x) \triangleq Pr(\mathbf{X} = x)$$

- When you see $p(x)$, it denotes the probability that a random variable X takes on the value x.
- The same format generalizes to mutiple random variables. E.g., $p(x, y, z)$ is the probability that three random variables, X, Y, and Z take on values x, y, and z respectively, i.e., $\Pr(X = x, Y = y, Z = z)$.
- If the random variable is ambiguous, we will denote it outright, e.g., $p_Y(x)$ would be the probability that a random variable Y takes on the value x.

With this notation, and for this class, we avoid using the following confusing expressions:

- $\Pr(x)$ – the probability of a number (rather than an event).
- $p(X)$ – the probability of a random variable taking on a random variable (rather than a value)
- $p(X = x)$ – the probability of a random variable taking on an event ...?

Manipulating probabilities

Manipulating probability expressions is an important part of machine learning. To do so, we largely employ two rules of probability: (1) the law of total probability (or sum rule), and (2) the probability chain rule (or product rule).

- Law of total probability:

$$\begin{aligned} p(x) &= \sum_y p(x,y) && x, y \text{ discrete} \\ p(x) &= \int_y p(x,y)dy && x, y \text{ continuous} \end{aligned}$$

- Probability chain rule:

$$\begin{aligned} \Pr(\text{Event}_1, \text{Event}_2) &= \Pr(\text{Event}_1)\Pr(\text{Event}_2|\text{Event}_1) \\ &= \Pr(\text{Event}_2)\Pr(\text{Event}_1|\text{Event}_2) \end{aligned}$$

Probability chain rule

The probability chain rule is very useful for manipulating probability expressions.

- Intuitively, we think of the chain rule as breaking up the joint probability into a product probability.
- E.g., if I want to evaluate $p(x, y)$, I can do this in two ways:
  - I can first evaluate the probability that $X = x$, and now given that $X = x$, evaluate the probability that $Y = y$. Hence,

$$p(x, y) = p(x)p(y|x)$$

  - I can also evaluate the probability that $Y = y$, and now given that $Y = y$, evaluate the probability that $X = x$. Hence,

$$p(x, y) = p(y)p(x|y)$$

Probability chain rule (cont.)

- Note that Event_1 and Event_2 could encompass multiple random variables taking on values.
- E.g., consider

$$\begin{aligned} \text{Event}_1 &= \{W = w, X = x\} \\ \text{Event}_2 &= \{Y = y, Z = z\} \end{aligned}$$

Then the probability chain rule states:

$$\begin{aligned} p(w, x, y, z) &= p(w, x)p(y, z|w, x) \\ &= p(y, z)p(w, x|y, z) \end{aligned}$$

Further decomposing, we could arrive at all sorts of equivalent expressions.

$$\begin{aligned} p(w, x, y, z) &= p(w, x)p(y, z|w, x) \\ &= p(x)p(w|x)p(y, z|w, x) && \text{Applied to } p(w, x) \\ &= p(x)p(w|x)p(z|w, x)p(y|z, w, x) && \text{Applied to } p(y, z|w, x) \end{aligned}$$

Probability chain rule (cont.)

- A helpful intuition: any event that has been in front of the conditioning bar must be behind the conditioning bar for all other probability expressions.
- This intuition states: once I evaluate the probability of a random variable taking on a certain value, I now assume that random variable *is* that value, and evaluate the probability of the remaining events.
- More examples to make this concrete, with the probability $p(w, x, y, z)$.
  - Evaluate the probability that $X = x$, then evaluate the probability that $W = w, Y = y, Z = z$ given that $X = x$:

$$p(w, x, y, z) = p(x)p(w, y, z|x)$$

  - Evaluate the probability that $W = w$ and $Z = z$, then $Y = y$ given the prior conditions, then $X = x$ given the prior conditions:

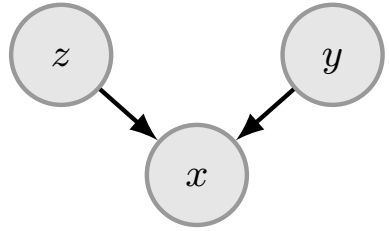
$$p(w, x, y, z) = p(w, z)p(y|w, z)p(x|w, z, y)$$

  - Evaluate the probability $W = w$, then $X = x$ given the prior condition, then $Y = y$, then $Z = z$:

$$p(w, x, y, z) = p(w)p(x|w)p(y|w, x)p(z|w, x, y)$$

Probability chain rule example

Graphical models represent conditional independencies in a succinct way. Though we don't cover them in class, here is a simple example:



This graph represents that x is influenced by z and y. This graph also has the following decomposition: $p(x, y, z) = p(z)p(y)p(x|y, z)$.

Example: Show that, for this model, $y \perp\!\!\!\perp z$.

From the chain rule of probability,

$$\begin{aligned} p(x, y, z) &= p(z)p(y|z)p(x|y, z) \\ &= p(y)p(z|y)p(x|y, z) \end{aligned}$$

Comparing these expressions to the model expression, we see that $p(z) = p(z|y)$ and $p(y) = p(y|z)$. Thus, y and z are independent.

Bayes' rule

By combining the probability chain rule and law of total probability, we can arrive at what is commonly called Bayes' rule or Bayes' theorem. The relationship is derived by recognizing:

$$\begin{aligned} p(x,y) &= p(x)p(y|x) \\ &= p(y)p(x|y) \end{aligned}$$

Hence,

$$\begin{aligned} p(x|y) &= \frac{p(y|x)p(x)}{p(y)} \\ &= \frac{p(y|x)p(x)}{\sum_x p(x,y)} \\ &= \frac{p(y|x)p(x)}{\sum_x p(y|x)p(x)} \end{aligned}$$

Thus, from Bayes rule, given $p(x)$ and $p(y|x)$, it is possible to calculate $p(x|y)$.

A word on Bayes' rule (10,000 foot level view of machine learning)

Bayes' rule appears frequently in machine learning, and is the basis for Bayesian inference. There's no expectation that you understand the details of this at the moment, but we can at least discuss intuition at a high-level.

Let x represent model parameters you wish to infer denoted θ, and let y correspond to data you observe denoted $\mathcal{D}$.

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\sum_x p(\mathcal{D}|\theta)p(\theta)}$$

What do each of these terms mean (in plain language)?

- $p(\theta|\mathcal{D})$ is the probability distribution of the model parameters given the data. This is sometimes called the *posterior* distribution.
- $p(\mathcal{D}|\theta)$ is the probability of having seen the data given a chosen set of model parameters. This is sometimes called the *likelihood* of the data.
- $p(\theta)$ are the probabilities of the model parameters absent of any data. This is sometimes called the *prior* on the parameters, since they represent the prior likelihood of each parameter before you saw any data. After you see data, the prior can be thought of as being "updated" by the likelihood to arrive at the posterior distribution on the parameters.

A word on Bayes' rule (cont.)

What can we do if we calculate these probabilities?

- In Bayesian inference, we calculate $p(\theta|\mathcal{D})$. Hence, we now have an *distribution* over the model parameters given the data we observed. We can then sample parameters from this distribution, or we can take the expected value of the parameters from the distribution, or the median value, etc. This concretely gives us all the parameters of our model.
- In a related area of machine learning, called maximum-likelihood estimation (or Frequentist inference), we instead focus on the $p(\mathcal{D}|\theta)$ term. We treat θ as a point (not a random variable with a distribution), and we want to infer the θ that makes the data most likely to have been observed. Hence, we choose the parameters that maximize the likelihood of the data, $p(\mathcal{D}|\theta)$.

There are benefits and cons of both approaches. Usually in this class, we'll do maximum-likelihood estimation. This doesn't communicate a preference for one or the other. Other machine learning classes will delve into these topics in more detail.

The need for derivatives

In machine learning, we often want to find the "best" model according to some performance metric. This language implies that we have to optimize the model to perform as well as possible.

Optimization research comprises a very large field, and even deep-learning specific optimization is something we'll discuss in detail in later weeks.

In optimization, derivatives are crucial. In simple quadratic examples from calculus, we could find minima and maxima exactly by setting the derivative equal to zero.

When the systems become more complex (e.g., with nonlinearities), there is no closed-form solution from setting the derivative equal to zero, but the derivative is still useful because it tells us how a change in the model parameters will affect our performance.

The scalar derivative

Consider a function $f : \mathbb{R} \to \mathbb{R}$ at a point $x \in \mathbb{R}$. Recall the definition of the derivative:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

The derivative tells us how much a small change in x changes f, i.e.,

$$f(x + \varepsilon) \approx f(x) + \varepsilon f'(x)$$

We denote $y = f(x)$ and denote the derivative of y with respect to x as $\frac{dy}{dx}$. Hence,

$$\Delta y \approx \frac{dy}{dx} \Delta x$$

The scalar chain rule

The scalar chain rule states that if $y = f(x)$ and $z = g(y)$, then

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

Intuitively: the chain rule tells us that a small change in x will cause a small change in y that will in turn cause a small change in z, i.e., for apropriately small Δx,

$$\begin{aligned} \Delta y &\approx \frac{dy}{dx}\Delta x \\ \Delta z &\approx \frac{dz}{dy}\Delta y \\ &= \frac{dz}{dy}\frac{dy}{dx}\Delta x \end{aligned}$$

A brief word on vector and matrix derivative notation

In this class, we will use both the differentiation operator ∂ and ∇ to denote derivatives. If y is a scalar, and $\mathbf{x}$ is a vector, then the gradient of y with respect to $\mathbf{x}$ is denoted as both:

$$\frac{\partial y}{\partial \mathbf{x}} \text{ and } \nabla_{\mathbf{x}} y$$

The gradient of y with respect to $\mathbf{x}$ is itself a vector with the same dimensionality as $\mathbf{x}$.

We use a similar notation for differentiation with respect to a matrix. Namely, if y is a scalar, and $\mathbf{A}$ is a matrix, then the derivative of y with respect to $\mathbf{A}$ is denoted as both:

$$\frac{\partial y}{\partial \mathbf{A}} \text{ and } \nabla_{\mathbf{A}} y$$

More on this later, but we use the "denominator layout" notation in this class, so that $\nabla_{\mathbf{A}} y$ has the same dimensionality as $\mathbf{A}$.

The gradient

The gradient generalizes the scalar derivative to multiple dimensions. Suppose $f : \mathbb{R}^n \to \mathbb{R}$ transforms a vector $\mathbf{x} \in \mathbb{R}^n$ to a scalar. If $y = f(\mathbf{x})$, then the gradient is:

$$\nabla_{\mathbf{x}} y = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix}$$

In other words, the gradient is:

- A *vector* that is the same size as $\mathbf{x}$, i.e., if $\mathbf{x} \in \mathbb{R}^n$ then $\nabla_{\mathbf{x}} y \in \mathbb{R}^n$.
- Each dimension of $\nabla_{\mathbf{x}} y$ tells us how small changes in $\mathbf{x}$ in that dimension affect y. i.e., changing the ith dimension of $\mathbf{x}$ by a small amount, Δx_i, will change y by

  $$\frac{\partial y}{\partial x_i} \Delta x_i$$

  We may also denote this as:

  $$\left(\nabla_{\mathbf{x}} y\right)_i \Delta x_i$$

Example: derivative with respect to a vector

Let $f(\mathbf{x}) = \theta^T \mathbf{x}$. What is $\nabla_{\mathbf{x}} f(\mathbf{x})$?

First, we note that $\theta^T \mathbf{x} = \sum_i \theta_i z_i$. Hence, $\frac{\partial f(\mathbf{x})}{\partial x_i} = \theta_i$. That is,

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

Example: derivative with respect to a vector

Let $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$. What is $\nabla_{\mathbf{x}} f(\mathbf{x})$?

First, we note that $\mathbf{x}^T\mathbf{A}\mathbf{x} = \sum_i \sum_j a_{ij}x_i x_j$. Then, we have

$$\begin{aligned}
\nabla_{\mathbf{x}} f(\mathbf{x}) &= \begin{bmatrix}
2a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + a_{21}x_2 + \cdots + a_{n1}x_n \\
2a_{22}x_2 + a_{21}x_1 + \cdots + a_{2n}x_n + a_{12}x_1 + \cdots + a_{n2}x_n \\
\vdots \\
2a_{nn}x_n + a_{n1}x_1 + \cdots + a_{n,n-1}x_{n-1} + a_{1n}x_1 + \cdots + a_{n-1,n}x_{n-1}
\end{bmatrix} \\
&= \mathbf{A}\mathbf{x} + \mathbf{A}^T\mathbf{x}
\end{aligned}$$

Note that if $\mathbf{A}$ is symmetric, then this further simplifies to:

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = 2\mathbf{A}\mathbf{x}$$

(Also note that if $\mathbf{A}$ and $\mathbf{x}$ are scalars, i.e., 1-dimensional, this is consistent with the scalar derivative.)

Derivative of a scalar w.r.t. a matrix

The derivative of a scalar, y, with respect to a matrix, $\mathbf{A} \in \mathbb{R}^{m \times n}$, is given by:

$$\nabla_{\mathbf{A}} y = \begin{bmatrix} \frac{\partial y}{\partial a_{11}} & \frac{\partial y}{\partial a_{12}} & \cdots & \frac{\partial y}{\partial a_{1n}} \\ \frac{\partial y}{\partial a_{21}} & \frac{\partial y}{\partial a_{22}} & \cdots & \frac{\partial y}{\partial a_{2n}} \\ \cdots & \cdots & \ddots & \vdots \\ \frac{\partial y}{\partial a_{m1}} & \frac{\partial y}{\partial a_{m2}} & \cdots & \frac{\partial y}{\partial a_{mn}} \end{bmatrix}$$

Like the gradient, the i, jth element of $\nabla_{\mathbf{A}} y$ tells us how small changes in a_{ij} affect y.

Note:

- If you search for the derivative of a scalar with respect to a matrix, you may find people give a transposed definition to the one above.
- Both are valid, but you must be consistent with your notation and use the correct rules. Our notation is called "denominator layout" notation; the other layout is called "numerator layout" notation.
- In the denominator layout, the dimensions of $\nabla_{\mathbf{A}} y$ and $\mathbf{A}$ are the same. The same holds for the gradient, i.e., the dimensions of $\nabla_{\mathbf{x}} y$ and $\mathbf{x}$ are the same. In the numerator layout notation, the dimensions are transposed.
- More on this later, but in "denominator layout," the chain rule goes right to left as opposed to left to right.

Derivative of a vector w.r.t. a vector

Let $\mathbf{y} \in \mathbb{R}^n$ be a function of $\mathbf{x} \in \mathbb{R}^m$. What dimensionality should the derivative of $\mathbf{y}$ with respect to $\mathbf{x}$ be?

- e.g., to see how $\Delta\mathbf{x}$ modifies y_i, we would calculate:

$$\Delta y_i = \nabla_{\mathbf{x}} y_i \cdot \Delta\mathbf{x}$$

- This suggests that the derivative ought to be an $n \times m$ matrix, denoted $\mathbf{J}$, of the form:

$$\mathbf{J} = \begin{bmatrix} (\nabla_{\mathbf{x}} y_1)^T \\ (\nabla_{\mathbf{x}} y_2)^T \\ \vdots \\ (\nabla_{\mathbf{x}} y_n)^T \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_m} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_m} \end{bmatrix}$$

The matrix would tell us how a small change in $\Delta\mathbf{x}$ results in a small change in $\Delta\mathbf{y}$ according to the formula:

$$\Delta\mathbf{y} \approx \mathbf{J}\Delta\mathbf{x}$$

Derivative of a vector w.r.t. a vector (cont.)

The matrix $\mathbf{J}$ is called the Jacobian matrix.

A word on notation:

- In the denominator layout definition, the denominator vector changes along rows.

$$\begin{aligned}\nabla_{\mathbf{x}}\mathbf{y} &= \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_m} & \frac{\partial y_2}{\partial x_m} & \cdots & \frac{\partial y_n}{\partial x_m} \end{bmatrix} \\ &\triangleq \frac{\partial \mathbf{y}}{\partial \mathbf{x}}\end{aligned}$$

- Hence the notation we use for the Jacobian would be:

$$\begin{aligned}\mathbf{J} &= (\nabla_{\mathbf{x}}\mathbf{y})^T \\ &= \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right)^T\end{aligned}$$

Example: derivative of a vector with respect to a vector

The following derivative will appear later on in the class. Let $\mathbf{W} \in \mathbb{R}^{h \times n}$ and $\mathbf{x} \in \mathbb{R}^n$. We would like to calculate the derivative of $f(\mathbf{x}) = \mathbf{W}\mathbf{x}$ with respect to $\mathbf{x}$.

$$\begin{aligned}
\nabla_{\mathbf{x}} \mathbf{W}\mathbf{x} &= \nabla_{\mathbf{x}} \begin{bmatrix} w_{11}x_1 + w_{12}x_2 + \cdots + w_{1n}x_n \\ w_{21}x_1 + w_{22}x_2 + \cdots + w_{2n}x_n \\ \vdots \\ w_{h1}x_1 + w_{h2}x_2 + \cdots + w_{hn}x_n \end{bmatrix} \\
&= \begin{bmatrix} w_{11} & w_{21} & \ldots & w_{h1} \\ w_{12} & w_{22} & \ldots & w_{h2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1n} & w_{2n} & \ldots & w_{hn} \end{bmatrix} \\
&= \mathbf{W}^T
\end{aligned}$$

Hessian

The Hessian matrix of a function $f(\mathbf{x})$ is a square matrix of second-order partial derivatives of $f(\mathbf{x})$. It is composed of elements:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial f}{\partial x_1^2} & \frac{\partial f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial f}{\partial x_1 \partial x_n} \\ \frac{\partial f}{\partial x_2 x_1} & \frac{\partial f}{\partial x_2^2} & \cdots & \frac{\partial f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_m x_1} & \frac{\partial f}{\partial x_m \partial x_2} & \cdots & \frac{\partial f}{\partial x_m^2} \end{bmatrix}$$

We can denote this matrix as $\nabla_{\mathbf{x}} (\nabla_{\mathbf{x}} f(\mathbf{x}))$. We often denote this simply as $\nabla_{\mathbf{x}}^2 f(\mathbf{x})$.

Chain rule for vector valued functions

In the "denominator" layout, the chain rule runs from right to left. We won't derive this, but we will check the dimensionality and intuition.

Let $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^n$, and $\mathbf{z} \in \mathbb{R}^p$. Further, let $\mathbf{y} = f(\mathbf{x})$ for $f : \mathbb{R}^m \to \mathbb{R}^n$ and $\mathbf{z} = g(\mathbf{y})$ for $g : \mathbb{R}^n \to \mathbb{R}^p$. Then,

$$\nabla_{\mathbf{x}} \mathbf{z} \quad = \quad \nabla_{\mathbf{x}} \mathbf{y} \, \nabla_{\mathbf{y}} \mathbf{z}$$

Equivalently:

$$\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \quad = \quad \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \frac{\partial \mathbf{z}}{\partial \mathbf{y}}$$

- Note that $\nabla_{\mathbf{x}} \mathbf{z}$ should have dimensionality $\mathbb{R}^{m \times p}$.
- As $\nabla_{\mathbf{x}} \mathbf{y} \in \mathbb{R}^{m \times n}$ and $\nabla_{\mathbf{y}} \mathbf{z} \in \mathbb{R}^{n \times p}$, the operations are dimension consistent.

Chain rule for vector valued functions (cont.)

- Intuitively, a small change $\Delta\mathbf{x}$ affects $\Delta\mathbf{z}$ through the Jacobian $(\nabla_{\mathbf{x}}\mathbf{z})^T$

$$\Delta\mathbf{z} \approx (\nabla_{\mathbf{x}}\mathbf{z})^T \Delta\mathbf{x} \quad (1)$$

- The chain rule is intuitive, since:

$$\begin{aligned} \Delta\mathbf{y} &\approx (\nabla_{\mathbf{x}}\mathbf{y})^T \Delta\mathbf{x} \\ \Delta\mathbf{z} &\approx (\nabla_{\mathbf{y}}\mathbf{z})^T \Delta\mathbf{y} \end{aligned}$$

  Composing these, we have that:

$$\Delta\mathbf{z} \approx (\nabla_{\mathbf{y}}\mathbf{z})^T (\nabla_{\mathbf{x}}\mathbf{y})^T \Delta\mathbf{x} \quad (2)$$

- Combining equations (1) and (2), we arrive at:

$$(\nabla_{\mathbf{x}}\mathbf{z})^T = (\nabla_{\mathbf{y}}\mathbf{z})^T (\nabla_{\mathbf{x}}\mathbf{y})^T$$

  which, after transposing both sides, reduces to the (right to left) chain rule:

$$\nabla_{\mathbf{x}}\mathbf{z} = \nabla_{\mathbf{x}}\mathbf{y}\, \nabla_{\mathbf{y}}\mathbf{z}$$

Derivatives of tensors

Occasionally in this class, we may need to take a derivative that is more than 2-dimensional. For example, we may want to take the derivative of a vector with respect to a matrix. This would be a 3-dimensional tensor. The definition for this would be as you expect. In particular, if $\mathbf{z} \in \mathbb{R}^p$ and $\mathbf{W} \in \mathbb{R}^{m \times n}$, then $\nabla_{\mathbf{W}} \mathbf{z}$ is a three-dimensional tensor with shape $\mathbb{R}^{m \times n \times p}$. Each $m \times n$ slice (of which there are p) is the matrix derivative $\nabla_{\mathbf{W}} z_i$.

Note, these are sometimes a headache to work with. We typically can find a shortcut to perform operations without having to compute and store these high-dimensional tensor derivatives. The next slides show an example.

Multivariate chain rule and tensor derivative example

Consider the squared loss function:

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{N}\left\|\mathbf{y}^{(i)} - \mathbf{W}\mathbf{x}^{(i)}\right\|^2$$

Here, $\mathbf{y}^{(i)} \in \mathbb{R}^m$ and $\mathbf{x}^{(i)} \in \mathbb{R}^n$ so that $\mathbf{W} \in \mathbb{R}^{m\times n}$. We wish to find $\mathbf{W}$ that minimizes the mean-square error in linearly predicting $\mathbf{y}^{(i)}$ from $\mathbf{x}^{(i)}$.

We consider one example, $\varepsilon^{(i)} = \frac{1}{2}\left\|\mathbf{y}^{(i)} - \mathbf{W}\mathbf{x}^{(i)}\right\|^2$.

We wish to calculate $\nabla_{\mathbf{W}}\varepsilon^{(i)}$.

To do so, we define $\mathbf{z}^{(i)} = \mathbf{y}^{(i)} - \mathbf{W}\mathbf{x}^{(i)}$. Then, $\varepsilon^{(i)} = \frac{1}{2}(\mathbf{z}^{(i)})^T\mathbf{z}^{(i)}$. For the rest of the example, we're going to drop the superscripts (i) and assume we're working with the ith example (to help notation). Then, we need to calculate is:

$$\nabla_{\mathbf{W}}\varepsilon = \nabla_{\mathbf{W}}\mathbf{z}\nabla_{\mathbf{z}}\varepsilon$$

Multivariate chain rule and tensor derivative example (cont.)

The first of these operations is straightforward. In particular,

$$\nabla_{\mathbf{z}}\varepsilon = \mathbf{z}$$

Now, we need to calculate $\nabla_{\mathbf{W}}\mathbf{z}$. This is a three dimensional tensor with dimensionality $m \times n \times m$.

This makes sense dimensionally, because when we multiply a $(m \times n \times m)$ tensor by a $(m \times 1)$ vector, we get out a $(m \times n \times 1)$ tensor, which is equivalently an $(m \times n)$ matrix. Because $\nabla_{\mathbf{W}}\varepsilon$ is an $(m \times n)$ matrix, this all works out.

Multivariate chain rule and tensor derivative example (cont.)

$\nabla_{\mathbf{W}}\mathbf{z}$ is an $(m \times n \times m)$ tensor.

- Letting z_k denote the kth element of $\mathbf{z}$, we see that each $\frac{\partial z_k}{\partial \mathbf{W}}$ is an $m \times n$ matrix, and that there are m of these for each element z_k from $k = 1, \ldots, m$.
- The *matrix*, $\frac{\partial z_k}{\partial \mathbf{W}}$, can be calculated as follows:

$$\frac{\partial z_k}{\partial \mathbf{W}} = \frac{\partial}{\partial \mathbf{W}} \sum_{j=1}^{n} -w_{kj} x_j$$

and thus, the (k, j)th element of this matrix is:

$$\left(\frac{\partial z_k}{\partial \mathbf{W}} \right)_{k,j} = \frac{\partial z_k}{\partial w_{kj}} = -x_j$$

It is worth noting that

$$\left(\frac{\partial z_k}{\partial \mathbf{W}} \right)_{i,j} = \frac{\partial z_k}{\partial w_{ij}} = 0 \qquad \text{for } k \neq i$$

Multivariate chain rule and tensor derivative example (cont.)

Hence, $\frac{\partial z_k}{\partial \mathbf{W}}$ is a matrix where the kth row is $\mathbf{x}^T$ and all other rows are the zeros (we denote the zero vector by $\mathbf{0}$). i.e.,

$$\frac{\partial z_1}{\partial \mathbf{W}} = \begin{bmatrix} -\mathbf{x}^T \\ \mathbf{0}^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} \qquad \frac{\partial z_2}{\partial \mathbf{W}} = \begin{bmatrix} \mathbf{0}^T \\ -\mathbf{x}^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} \qquad \text{etc...}$$

Now applying the chain rule,

$$\frac{\partial \varepsilon}{\partial \mathbf{W}} = \frac{\partial \mathbf{z}}{\partial \mathbf{W}} \frac{\partial \varepsilon}{\partial \mathbf{z}}$$

is a tensor product between an $(m \times n \times m)$ tensor and an $(m \times 1)$ vector, whose resulting dimensionality is $(m \times n \times 1)$ or equivalently, an $(m \times n)$ matrix.

Multivariate chain rule and tensor derivative example (cont.)

We carry out this tensor-vector multiply in the standard way.

$$\begin{aligned}
\frac{\partial \varepsilon}{\partial \mathbf{W}} &= \frac{\partial \mathbf{z}}{\partial \mathbf{W}} \frac{\partial \varepsilon}{\partial \mathbf{z}} \\
&= \sum_{i=1}^{m} \frac{\partial z_i}{\partial \mathbf{W}} \left(\frac{\partial \varepsilon}{\partial \mathbf{z}} \right)_i \\
&= \frac{\partial \varepsilon}{\partial z_1} \begin{bmatrix} -\mathbf{x}^T \\ \mathbf{0}^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} + \frac{\partial \varepsilon}{\partial z_2} \begin{bmatrix} \mathbf{0}^T \\ -\mathbf{x}^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} + \cdots \\
&\quad + \frac{\partial \varepsilon}{\partial z_m} \begin{bmatrix} \mathbf{0}^T \\ \mathbf{0}^T \\ \vdots \\ -\mathbf{x}^T \end{bmatrix}
\end{aligned}$$

Multivariate chain rule and tensor derivative example (cont.)

Continuing the work from the previous page...

$$\begin{aligned}\frac{\partial \varepsilon}{\partial \mathbf{W}} &= -\begin{bmatrix} \frac{\partial \varepsilon}{\partial z_1}\mathbf{x}^T \\ \frac{\partial \varepsilon}{\partial z_2}\mathbf{x}^T \\ \vdots \\ \frac{\partial \varepsilon}{\partial z_m}\mathbf{x}^T \end{bmatrix} \\ &= -\frac{\partial \varepsilon}{\partial \mathbf{z}}\mathbf{x}^T\end{aligned}$$

Hence, with a final application of the chain rule, we get that

$$\begin{aligned}\nabla_{\mathbf{W}}\varepsilon &= -\mathbf{z}\mathbf{x}^T \\ &= -(\mathbf{y} - \mathbf{W}\mathbf{x})\mathbf{x}^T\end{aligned}$$

Setting this equal to zero, we find that for one example,

$$\mathbf{W} = \mathbf{y}\mathbf{x}^T(\mathbf{x}\mathbf{x}^T)^{-1}$$

Summing across all examples, this produces least-squares.

A few notes on tensor derivatives

- In general, the simpler rule can be inferred via pattern intuition / looking at the dimensionality of the matrices, and these tensor derivatives need not be explicitly derived.
- Indeed, actually calculating these tensor derivatives, storing them, and then doing e.g., a tensor-vector multiply, is usually not a good idea for both memory and computation. In this example, storing all these zeros and performing the multiplications is unnecessary.
- If we know the end result is simply an outer product of two vectors, we need not even calculate an additional derivative in this step of backpropagation, or store an extra value (assuming the inputs were previously cached).