

Section 7

Admin

- Office Hours: Wednesday 1-2PM!
- Homework 2 thoughts?
- Homework 3: **March 2nd**
- Course and Section Feedback?

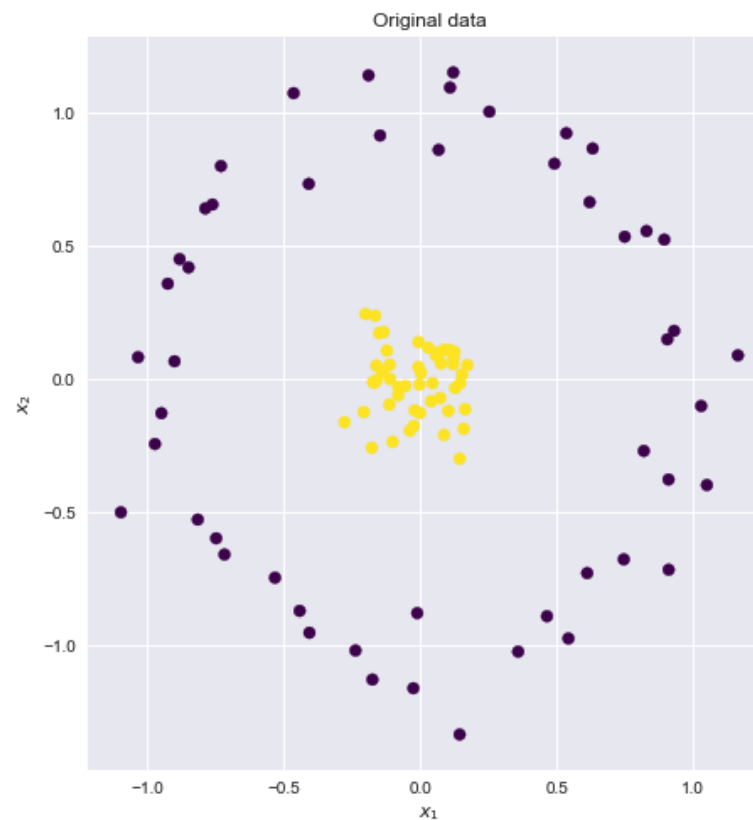
Agenda

- Kernels (~35 min)
 - Questions 1, 2, 3
- PyTorch Neural Network Edition (~15 min)
 - Notebook

Kernels

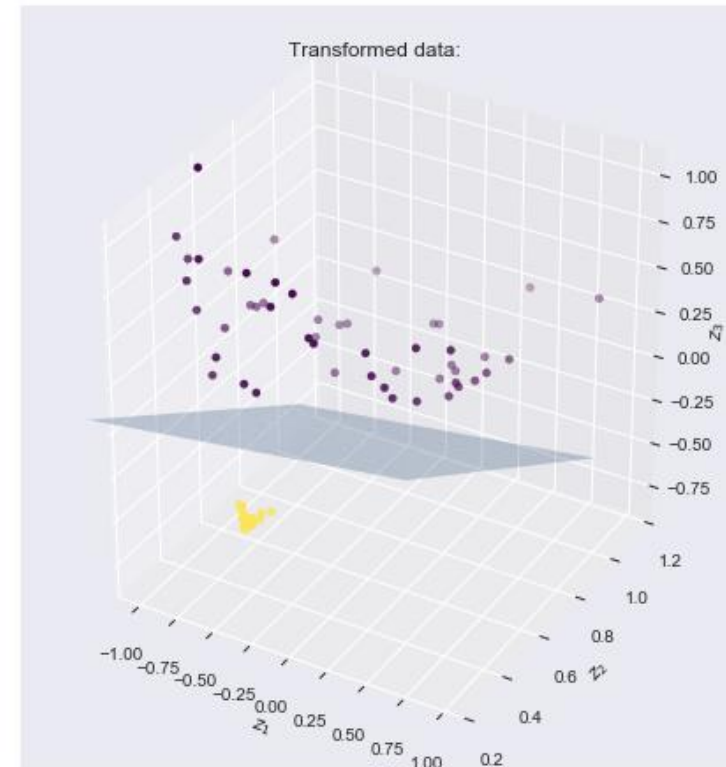
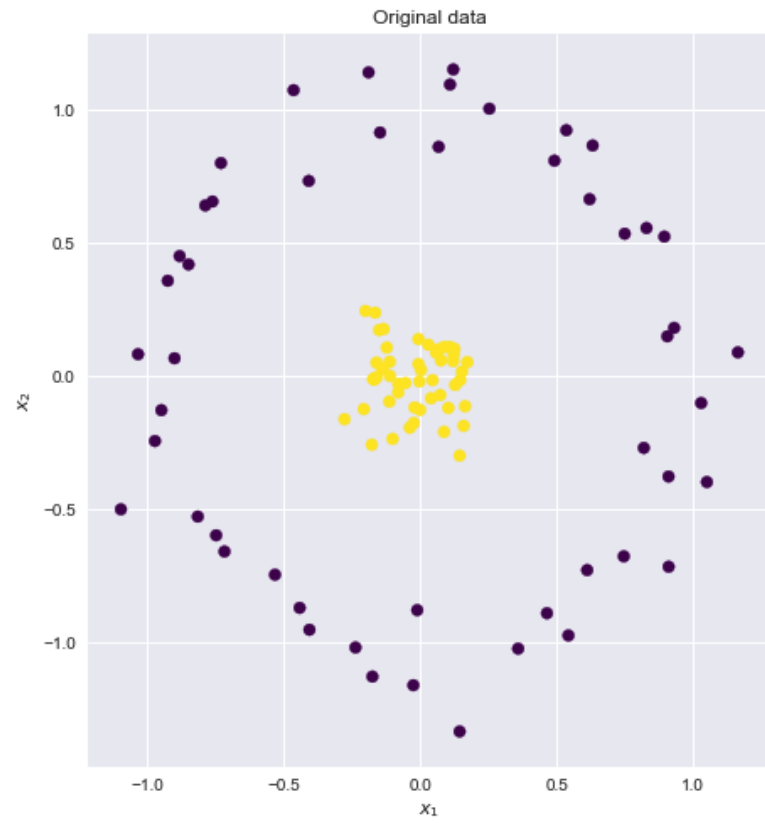
Kernels – Motivation

- Suppose we had input data and wanted to classify with a linear classifier (SVM)
- Clearly not linearly separable!
- What's the solution?



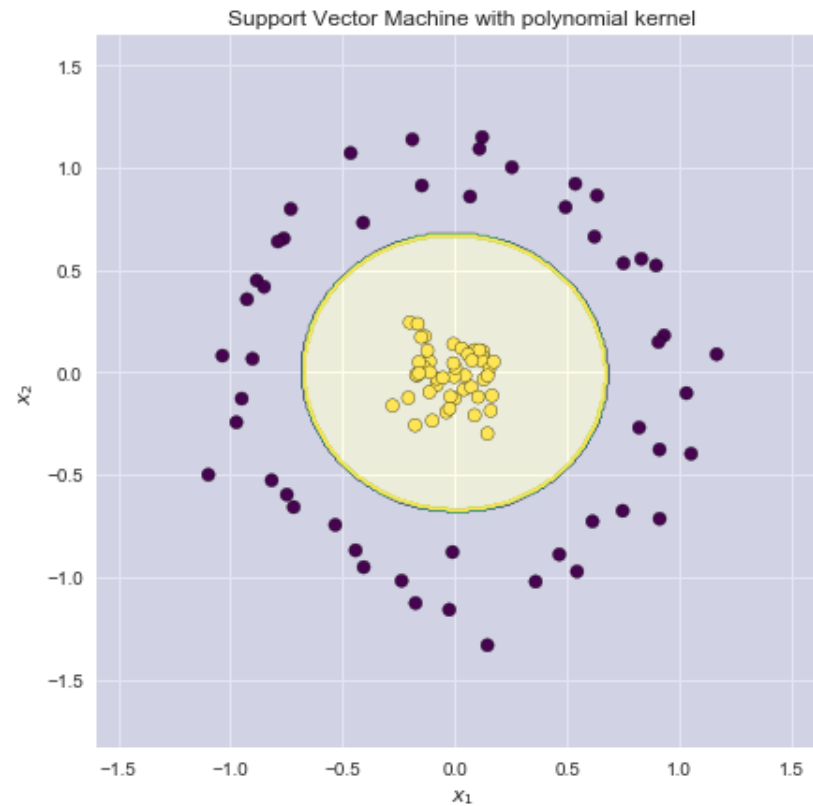
Kernels – Motivation

- Using a Kernel (polynomial):



Kernels – Motivation

- Using a Kernel (polynomial):



Kernels – Math

- **Definition:** Let $\Phi: \mathbb{R}^k \rightarrow \mathbb{R}^d$, then a function $K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a kernel for Φ if $K(x, x') = \phi(x) \cdot \phi(x')$ for all x, x' .
 - Can be thought of as a similarity metric for objects x, x' .

Kernels – Math

- **Definition:** Let $\Phi: \mathbb{R}^k \rightarrow \mathbb{R}^d$, then a function $K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a kernel for Φ if $K(x, x') = \phi(x) \cdot \phi(x')$ for all x, x' .
 - Can be thought of as a similarity metric for objects x, x' .

We've justified why high dimensional data is important, but why is a kernel helpful?

Kernels – Math Intuition

Given dataset $x = (1, 2, 3)$, $y = (3, 4, 5)$, and $\phi: z \rightarrow z \times z$, lets try and calculate $\langle \Phi(x), \Phi(y) \rangle$.

Kernels – Math Intuition

Given dataset $x = (1, 2, 3)$, $y = (3, 4, 5)$, and $\phi: z \rightarrow z \times z$, lets try and calculate $\langle \Phi(x), \Phi(y) \rangle$. Note that $K(x, y) = (\langle x, y \rangle)^2$.

Method 1:

$$\Phi(z) = \langle z_1z_1, z_1z_2, z_1z_3, z_2z_1, z_2z_2, z_2z_3, z_3z_1, z_3z_2, z_3z_3 \rangle$$

$$\Phi(x) = \langle 1, 2, 3, 2, 4, 6, 3, 6, 9 \rangle$$

$$\Phi(y) = \langle 16, 20, 24, 20, 25, 30, 24, 30, 36 \rangle$$

$$\langle \Phi(x), \Phi(y) \rangle = 16 + 40 + 72 + 40 + 100 + 180 + 72 + 180 + 324 = 1024$$

Method 2:

$$K(x, y) = (\langle x, y \rangle)^2 = (4 + 10 + 18)^2 = 1024$$

Which was easier?

Kernels – Kernelized Regression (1)

Consider regularized linear regression (without a bias, for simplicity). We want to find the optimal parameters $\hat{w} = \arg \min_w L(w)$ for k -featured dataset $(x_i, y_i)_{i=1}^n$ (i.e. $X \in \mathbb{R}^{n \times k}$) that minimizes the following loss function:

$$L(w) = \sum_{i=1}^n (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$$

From class, we know there is an optimal \hat{w} that lies in the span of the datapoints. Concretely, there exists $\alpha_1, \dots, \alpha_n \in \mathbb{R}$ such that $\hat{w} = \sum_{i=1}^n \alpha_i x_i$.

Kernels – Kernelized Regression (1a)

Let $\alpha \in \mathbb{R}^n$, $\mathbf{K} \in \mathbb{R}^{n \times n}$, $\mathbf{y} \in \mathbb{R}^n$. Further, let us assume that we are using a linear kernel where $\mathbf{K}_{ij} = x_i^T x_j$. Express the loss function $L(w)$ in terms of \mathbf{K} and α as $L(\alpha)$.

$$L(w) = \sum_{i=1}^n (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$$

$$\hat{w} = \sum_i^n \alpha_i x_i.$$

Kernels – Kernelized Regression (1a)

Let $\alpha \in \mathbb{R}^n$, $\mathbf{K} \in \mathbb{R}^{n \times n}$, $\mathbf{y} \in \mathbb{R}^n$. Further, let us assume that we are using a linear kernel where $\mathbf{K}_{ij} = x_i^T x_j$. Express the loss function $L(w)$ in terms of \mathbf{K} and α as $L(\alpha)$.

$$\begin{aligned} L(\alpha) &= \sum_i^n \left(\sum_j^n (\alpha_j x_j^T) x_i - y_i \right)^2 + \lambda \left\| \sum_i^n \alpha_i x_i \right\|^2 \\ &= \sum_i^n \left(\sum_j^n (\alpha_j x_j^T) x_i - y_i \right)^2 + \lambda \left(\sum_i^n \alpha_i x_i \right)^T \left(\sum_i^n \alpha_i x_i \right) \\ &= \sum_i^n \left(\sum_j^n \alpha_j \langle x_j, x_i \rangle - y_i \right)^2 + \lambda \sum_i^n \sum_j^n \alpha_i \alpha_j x_i^T x_j \\ &= \sum_i^n \left(\sum_j^n \alpha_j K(x_j, x_i) - y_i \right)^2 + \lambda \sum_i^n \sum_j^n \alpha_i \alpha_j K(x_i, x_j) \\ &= \|\mathbf{y} - \mathbf{K}\alpha\|_2^2 + \lambda \alpha^T \mathbf{K} \alpha \end{aligned}$$

Kernels – Kernelized Regression (1b)

Assuming that \mathbf{K} is invertible, solve for the optimal $\hat{\alpha}$.

$$\|\mathbf{y} - \mathbf{K}\alpha\|_2^2 + \lambda\alpha^T \mathbf{K}\alpha$$

Kernels – Kernelized Regression (1b)

Assuming that \mathbf{K} is invertible, solve for the optimal $\hat{\alpha}$. $\|\mathbf{y} - \mathbf{K}\alpha\|_2^2 + \lambda\alpha^T \mathbf{K}\alpha$

Set the gradient of $L(\alpha) = 0$:

$$\nabla L(\alpha) = 0$$

So then,

$$-2\mathbf{K}(\mathbf{y} - \mathbf{K}\alpha) + 2\lambda\mathbf{K}\alpha = 0$$

$$-\mathbf{K}(\mathbf{y} - \mathbf{K}\alpha) + \lambda\mathbf{K}\alpha = 0$$

$$\mathbf{K}(\mathbf{K}\alpha - \mathbf{y} + \lambda\alpha) = 0$$

$$\mathbf{K}((\mathbf{K} + \lambda I)\alpha - \mathbf{y}) = 0$$

$$\mathbf{K}(\mathbf{K} + \lambda I)\alpha = \mathbf{K}\mathbf{y}$$

$$\hat{\alpha} = (\mathbf{K} + \lambda I)^{-1}\mathbf{y}$$

Kernels – Kernelized Regression (1c)

Suppose after training our model on $\mathbf{X}_{\text{train}}$ we seek to make predictions on \mathbf{X}_{test} . Express these predictions $\hat{\mathbf{Y}}$ in terms of $\mathbf{K}_{\text{train}} = \mathbf{X}_{\text{train}}\mathbf{X}_{\text{train}}^T$, $\mathbf{y}_{\text{train}}$, $\mathbf{X}_{\text{train}}$, and \mathbf{X}_{test} . What would the general prediction formula look like if we are not using a linear kernel? Express the solution in terms of $\mathbf{K}_{\text{train, test}} = \Phi(\mathbf{X}_{\text{test}})\Phi(\mathbf{X}_{\text{train}}^T)$ and $\hat{\alpha}$, where for arbitrary d , we have $\Phi : \mathbb{R}^k \rightarrow \mathbb{R}^d$.

Kernels – Kernelized Regression (1c)

Suppose after training our model on $\mathbf{X}_{\text{train}}$ we seek to make predictions on \mathbf{X}_{test} . Express these predictions $\hat{\mathbf{Y}}$ in terms of $\mathbf{K}_{\text{train}} = \mathbf{X}_{\text{train}}\mathbf{X}_{\text{train}}^T$, $\mathbf{y}_{\text{train}}$, $\mathbf{X}_{\text{train}}$, and \mathbf{X}_{test} . What would the general prediction formula look like if we are not using a linear kernel? Express the solution in terms of $\mathbf{K}_{\text{train, test}} = \Phi(\mathbf{X}_{\text{test}})\Phi(\mathbf{X}_{\text{train}}^T)$ and $\hat{\alpha}$, where for arbitrary d , we have $\Phi : \mathbb{R}^k \rightarrow \mathbb{R}^d$.

$$\begin{aligned}\hat{\mathbf{Y}} &= \mathbf{X}_{\text{test}}\hat{\mathbf{w}} \\ &= \mathbf{X}_{\text{test}}\mathbf{X}_{\text{Train}}^T\hat{\alpha} \\ &= \mathbf{X}_{\text{test}}\mathbf{X}_{\text{train}}^T(\mathbf{K}_{\text{train}} + \lambda I)^{-1}\mathbf{y}_{\text{train}}\end{aligned}$$

General Solution for Kernel Ridge

$$\begin{aligned}\hat{\mathbf{Y}} &= \Phi(\mathbf{X}_{\text{test}})\Phi(\mathbf{X}_{\text{train}}^T)(\mathbf{K}_{\text{train}} + \lambda I)^{-1}\mathbf{y}_{\text{train}} \\ &= \mathbf{K}_{\text{train, test}}\hat{\alpha}\end{aligned}$$

Kernels – Proving $\hat{w} \in \text{Span}(x_1, \dots, x_n)$ (2)

Why is this something we need to show?

Kernels – Proving $\hat{w} \in \text{Span}(x_1, \dots, x_n)$ (2)

Why is this something we need to show?

We will prove this through contradiction. Assume $\hat{w} \notin \text{Span}(x_1, \dots, x_n)$ solves $\arg \min_w L(w)$ where $L(w)$ is defined above. Then, there exists a component of \hat{w} that is perpendicular to the span, which we will call w^\perp . Concretely,

$$\hat{w} = \bar{w} + w^\perp$$

Where $\bar{w} = \sum_{i=1}^n \alpha_i x_i$ is the component of \hat{w} in the span of the datapoints.

Kernels – Proving $\hat{w} \in \text{Span}(x_1, \dots, x_n)$ (2a)

Show that $\hat{w} \cdot x_i = \bar{w} \cdot x_i$, for every x_i . (Hint: what is the relationship of w^\perp and x_i)

Kernels – Proving $\hat{w} \in \text{Span}(x_1, \dots, x_n)$ (2a)

Show that $\hat{w} \cdot x_i = \bar{w} \cdot x_i$, for every x_i . (Hint: what is the relationship of w^\perp and x_i)

$$\begin{aligned}\hat{w} \cdot x_i &= (\bar{w} + w^\perp) \cdot x_i \\ &= \bar{w} \cdot x_i + w^\perp \cdot x_i \\ &= \bar{w} \cdot x_i + 0 \\ &= \bar{w} \cdot x_i\end{aligned}$$

w^\perp is perpendicular to each x_i

Kernels – Proving $\hat{w} \in \text{Span}(x_1, \dots, x_n)$ (2b)

Now, show that $\|\hat{w}\|_2^2 \geq \|\bar{w}\|_2^2$. (Hint: Use the Pythagorean Theorem)

Kernels – Proving $\hat{w} \in \text{Span}(x_1, \dots, x_n)$ (2b)

Now, show that $\|\hat{w}\|_2^2 \geq \|\bar{w}\|_2^2$. (Hint: Use the Pythagorean Theorem)

$$\begin{aligned}\|\hat{w}\|_2^2 &= \|\bar{w} + w^\perp\|_2^2 \\ &= \|\bar{w}\|_2^2 + \|w^\perp\|_2^2 && \text{Pythagorean Theorem} \\ &\geq \|\bar{w}\|_2^2\end{aligned}$$

Kernels – Proving $\hat{w} \in \text{Span}(x_1, \dots, x_n)$ (2c)

Finally, show that $\hat{w} \in \text{Span}(x_1, \dots, x_n)$. (Hint: Think about the regularization term)

Kernels – Proving $\hat{w} \in \text{Span}(x_1, \dots, x_n)$ (2c)

Finally, show that $\hat{w} \in \text{Span}(x_1, \dots, x_n)$. (Hint: Think about the regularization term)

Note that in the loss function we're trying to minimize the magnitude of w (with the regularization term $\lambda \|w\|_2^2$). Note that if $\forall_i \hat{w}^T x_i = \bar{w}^T x_i$, and $\|\hat{w}\|_2^2 \geq \|\bar{w}\|_2^2$, then our optimization will always choose $w^\perp = 0$, meaning that $\hat{w} = \bar{w}$ and $\hat{w} \in \text{Span}(x_1, \dots, x_n)$, which completes the contradiction.

Kernels – Kernel Proofs (3a)

Let $\phi : d \rightarrow k$ be a feature map, and define K to be the kernel matrix of ϕ . Prove that the kernel matrix is symmetric. That is, show $K_{i,j} = K_{j,i}$.

Kernels – Kernel Proofs (3a)

Let $\phi : d \rightarrow k$ be a feature map, and define K to be the kernel matrix of ϕ .

Prove that the kernel matrix is symmetric. That is, show $K_{i,j} = K_{j,i}$.

Let $\phi(x_i)$ and $\phi(x_j)$ be the feature maps for x_i and x_j , respectively. Then $K_{i,j} = \phi(x_i)^T \phi(x_j) = \phi(x_j)^T \phi(x_i) = K_{j,i}$.

Kernels – Kernel Proofs (3b)

Let $\phi : d \rightarrow k$ be a feature map, and define K to be the kernel matrix of ϕ .

Recall that a matrix M is positive semi-definite if $x^T M x \geq 0, \forall x \in \mathbb{R}^n$.

Show that K is positive semi-definite. (Hint: consider the matrix B where the i^{th} row of B is $\phi(x_i)$).

Kernels – Kernel Proofs (3b)

Let $\phi : d \rightarrow k$ be a feature map, and define K to be the kernel matrix of ϕ .

Recall that a matrix M is positive semi-definite if $x^T M x \geq 0, \forall x \in \mathbb{R}^n$.

Show that K is positive semi-definite. (Hint: consider the matrix B where the i^{th} column of B is $\phi(x_i)$).

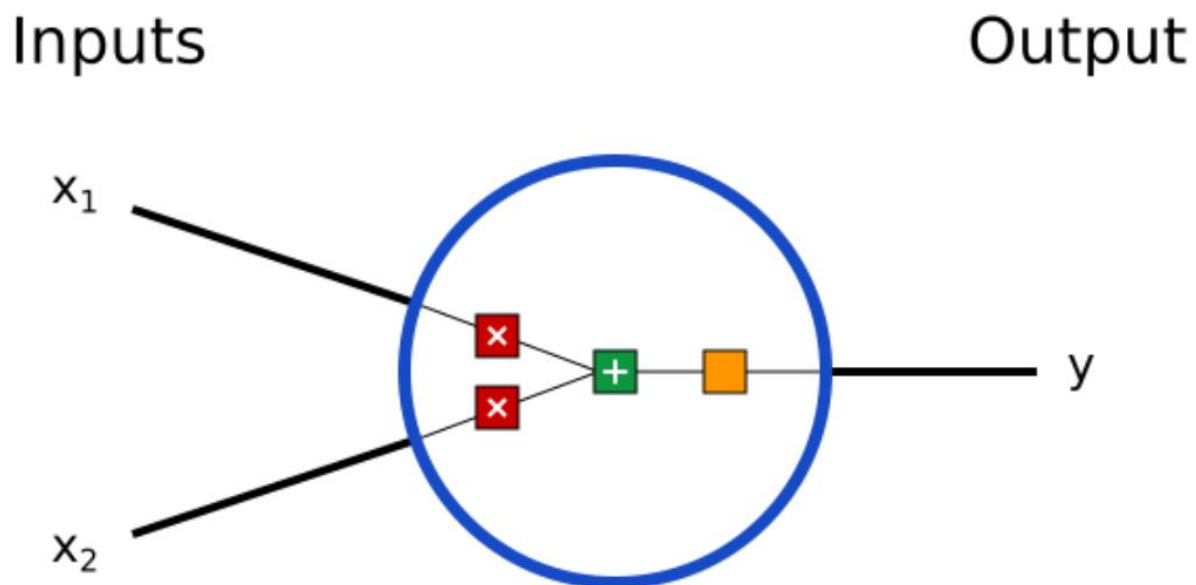
Recall that $K_{i,j} = \phi(x_i)^T \phi(x_j)$. Observe that $K = B^T B$, as $(B^T B)_{i,j} = \phi(x_i)^T \phi(x_j)$. Now consider an arbitrary vector y . To show K is PSD it suffices to show $y^T K y$ is non-negative. We have:

$$y^T K y = y^T B^T B y = (B y)^T (B y) = \|B y\|_2^2 \geq 0$$

PyTorch Neural Network Edition

Neural Networks – What is a Neuron?

- Basic unit of a neural network
- Takes an input, does some math, creates a single output



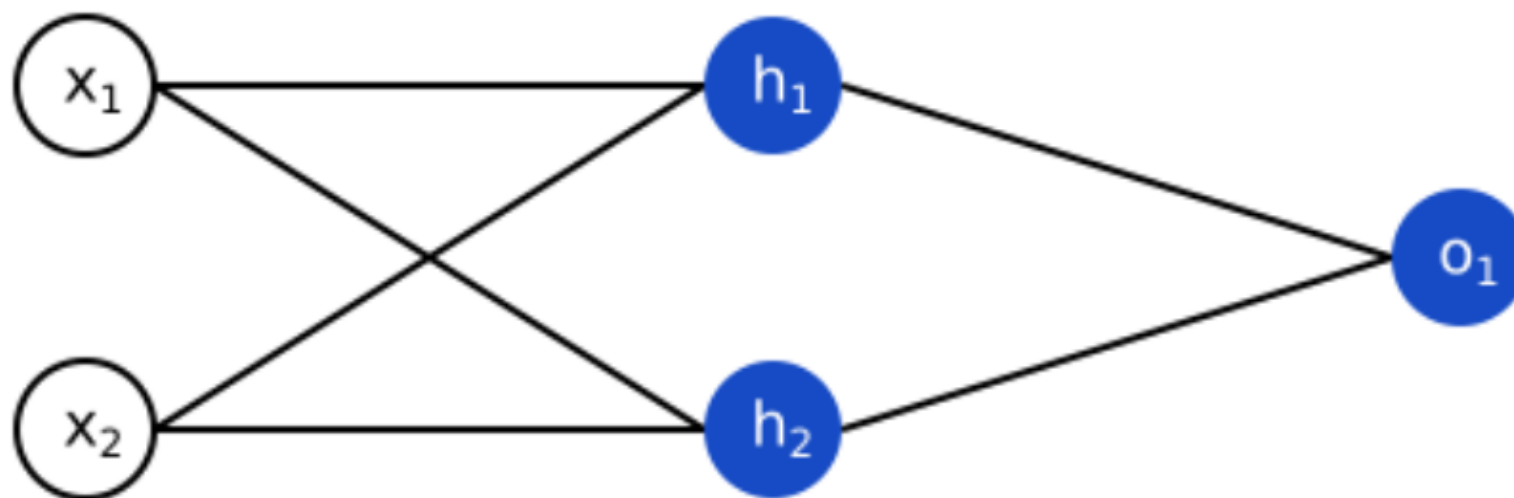
Neural Networks – What is a Layer?

- A collection of neurons that return a collection of outputs

Input Layer

Hidden Layer

Output Layer



Neural Networks – Types of Layers

- Fully-Connected
- Convolutional
- Attention
- ... and many more!

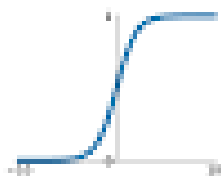
Neural Networks – What is activation?

- A function to apply to the output of the layer
- Has important geometric results that we will discuss in class

Activation Functions

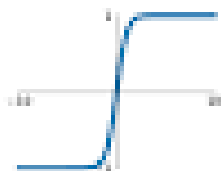
Sigmoid

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



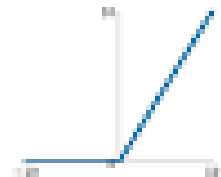
tanh

$$\tanh(x)$$



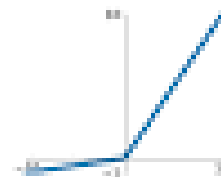
ReLU

$$\max(0, x)$$



Leaky ReLU

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

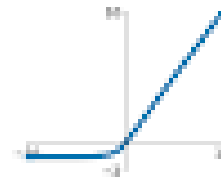


Maxout

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

ELU

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



Neural Networks – How do we train?

- An algorithm called backpropagation! **More in class.**

