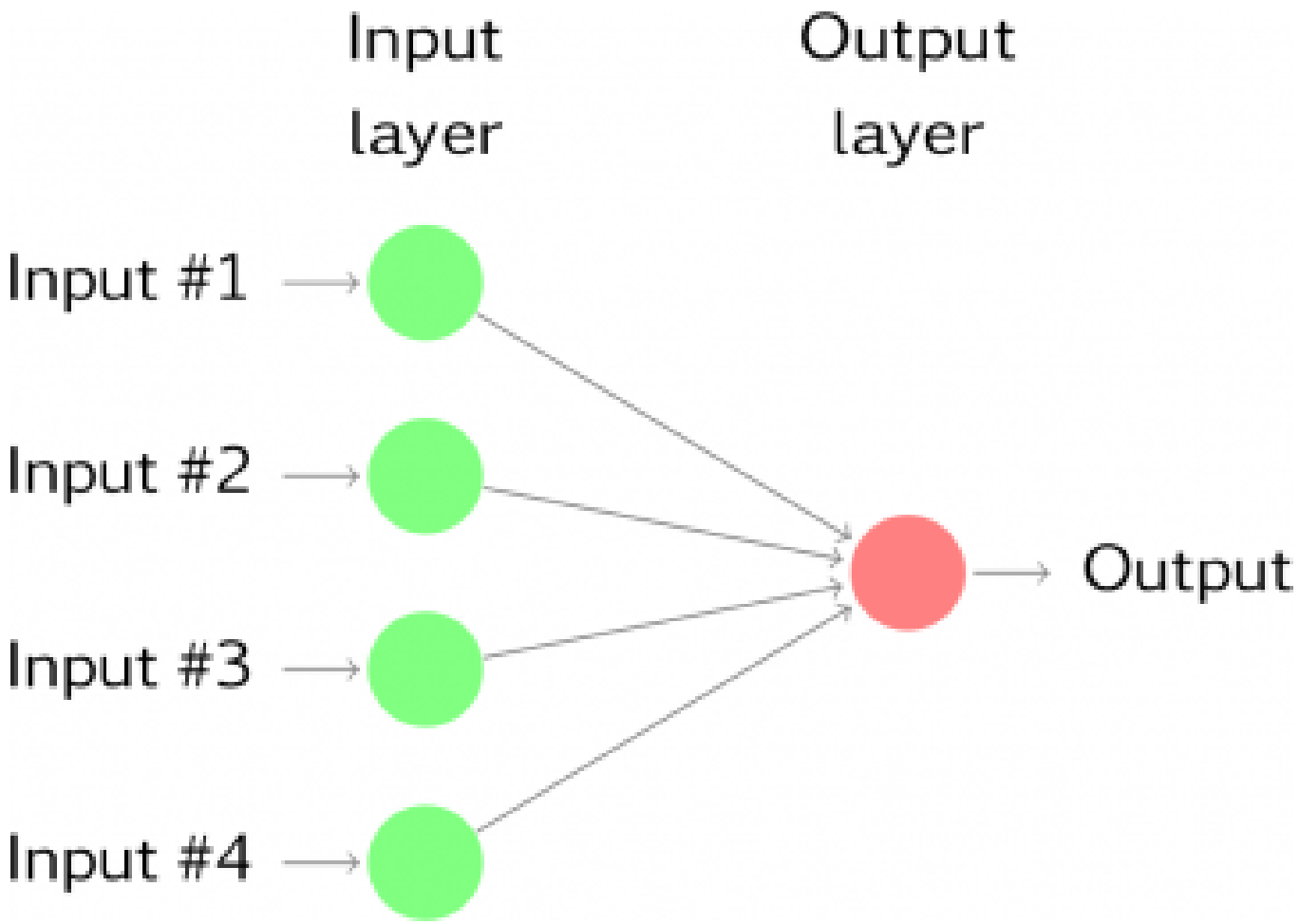
Introduction to Kernel Methods

Matt Galloway

March 30, 2018


Input
layer
Output
layer
Input #1
Input #2
Input #3
Input #4
Output

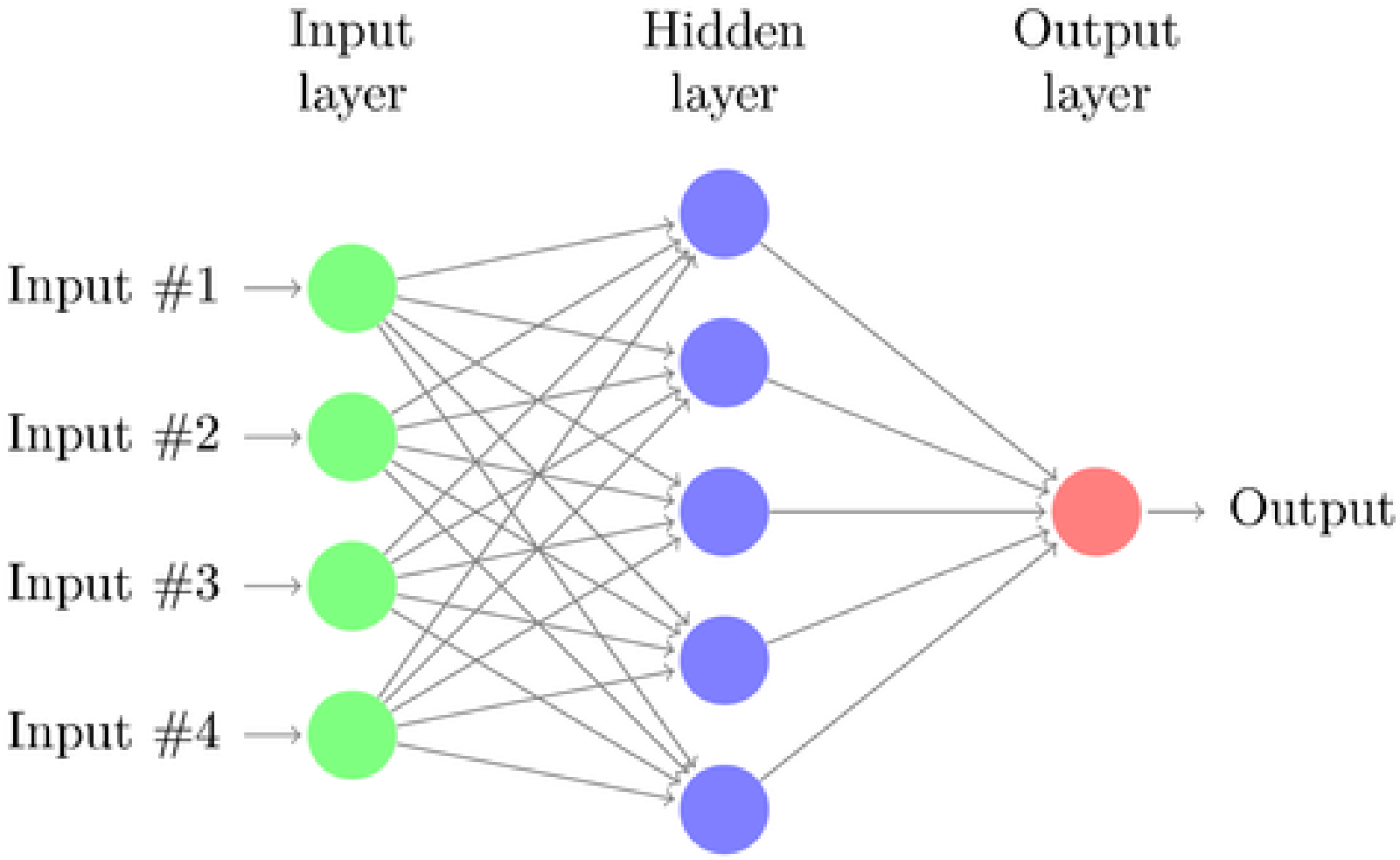
Motivation

- Linear model:

$$f(x) = x^T\beta = \beta_0 + \beta_1 x_1 + \beta x_2 + \epsilon$$

- Added polynomial and interaction terms:

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \epsilon$$


Input layer
Hidden layer
Output layer
Input #1
Input #2
Input #3
Input #4
Output

- Linear model:

$$f(x) = x^T\beta = \beta_0 + \beta_1 x_1 + \beta x_2 + \epsilon$$

- Added polynomial and interaction terms:

$$\begin{aligned} f(x) &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \epsilon \\ &= \tilde{\beta}_0 + \tilde{\beta}_1\sqrt{2}x_1 + \tilde{\beta}_2\sqrt{2}x_2 + \tilde{\beta}_3 x_1^2 + \tilde{\beta}_4 x_2^2 + \tilde{\beta}_5\sqrt{2}x_1 x_2 + \epsilon \\ &= \tilde{\beta}^T\phi(x) + \epsilon \end{aligned}$$

where ϕ is a mapping: $\phi : \mathbb{R}^2 \to \mathbb{R}^5$

$$\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1 x_2, x_2^2)^T$$

- We see that ϕ is a mapping into a larger dimension:

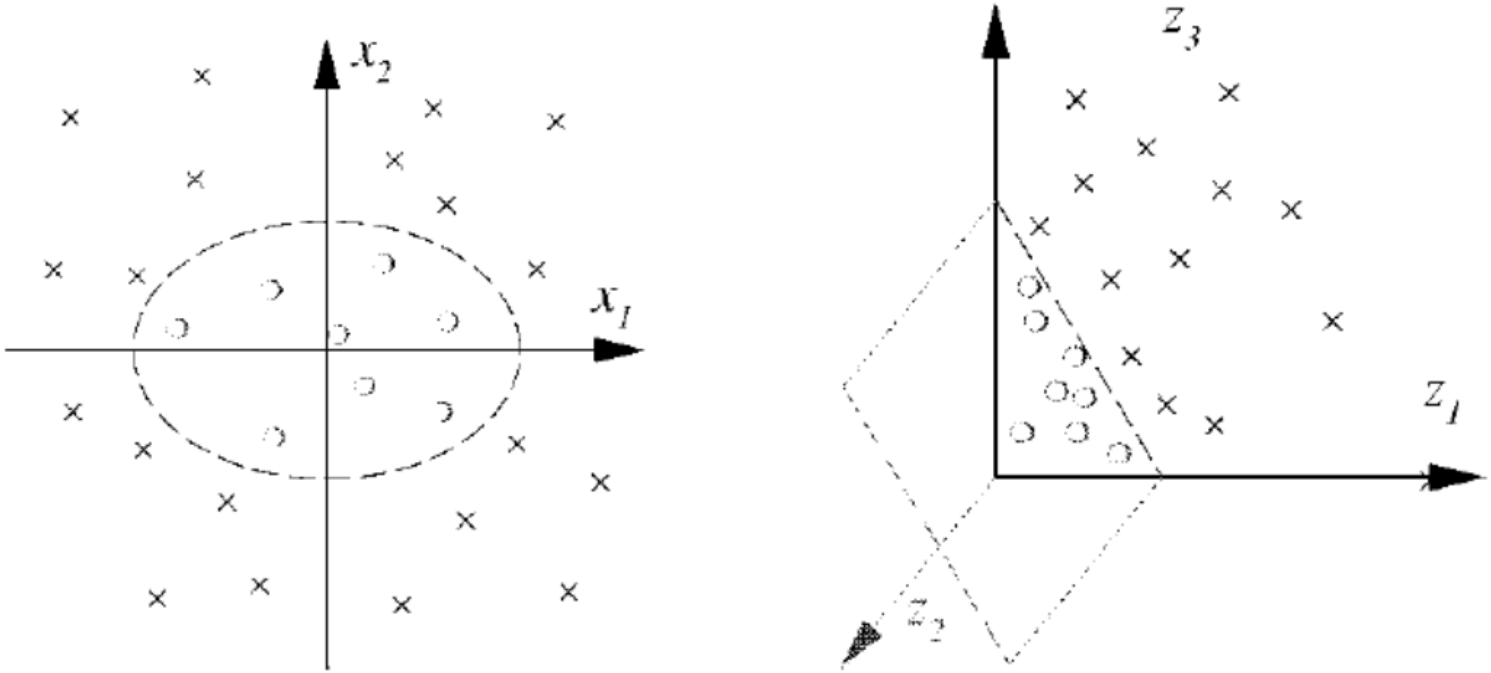
$$\phi : \mathbb{R}^2 \to \mathbb{R}^5$$

- In our case, include all polynomial terms up to order 2 and all interaction terms:

$$\phi(\begin{bmatrix} x_1 & x_2 \end{bmatrix}) = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 \end{bmatrix}$$

where $z_1 = \sqrt{2}x_1, z_2 = \sqrt{2}x_2, z_3 = x_1^4, z_4 = \sqrt{2}x_1x_2, x_5 = x_2^2$.

Motivation

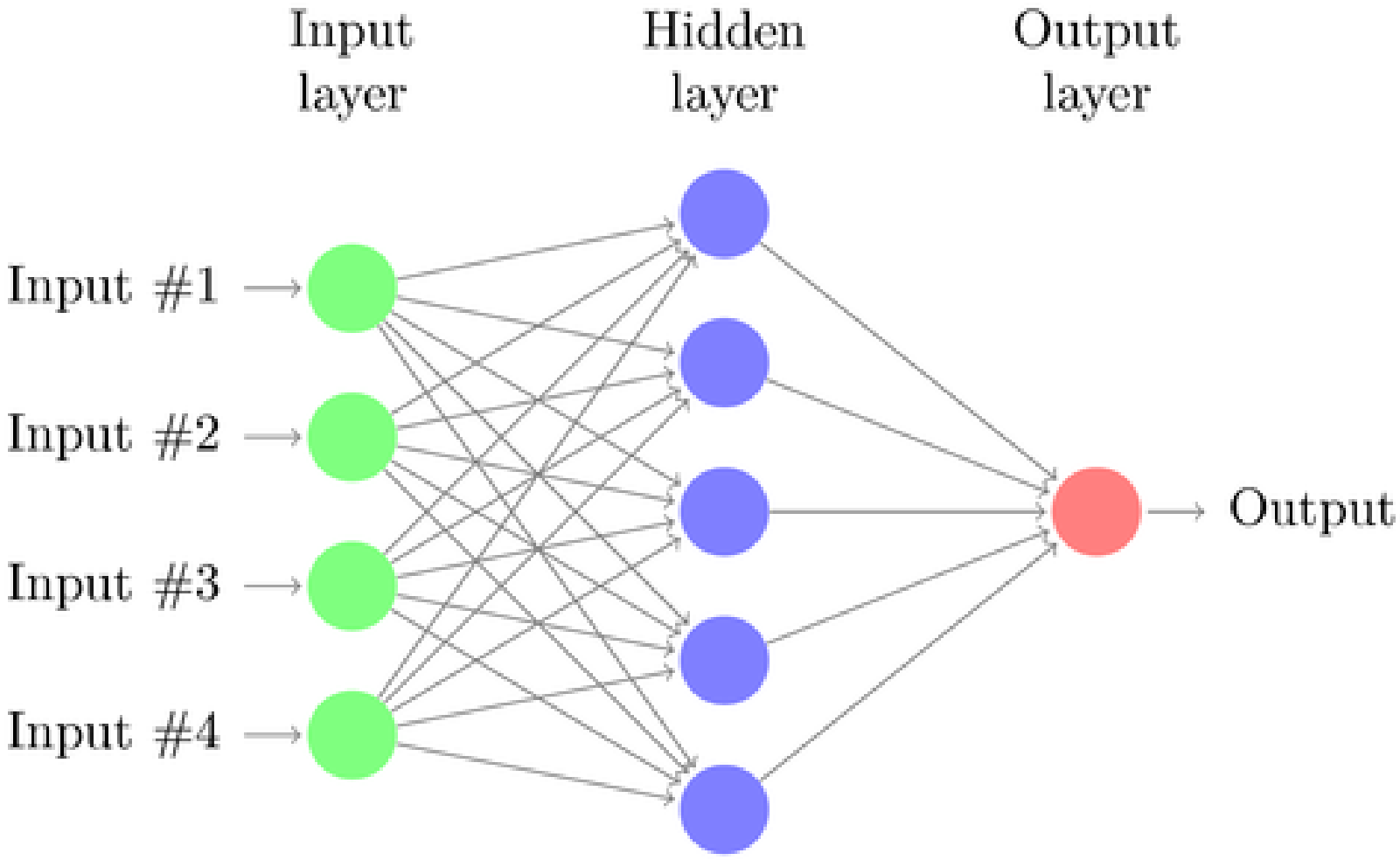


- What about when p is large (high dimensional setting)?

Example: Suppose $p = 100$ and we want to model **all** polynomial terms up to order 3 and all interaction terms:

$$\phi : \mathbb{R}^{100} \to \mathbb{R}^{BIG}$$

- The dimension of our data increases significantly!
- **Problem:** How do we make this procedure computationally feasible?


Input layer
Hidden layer
Output layer
Input #1
Input #2
Input #3
Input #4
Output

- Generic problem: for $\lambda \geq 0$,

$$\min_{\beta} L(X\beta) + \lambda \|\beta\|_2^2$$

- By the **Representer Theorem**, we can show that the optimal β (say β^*) lies in the span of the data points:

That is, $\beta^* = X^T v$ for some $v \in \mathbb{R}^n$:

$$\min_{\beta} L(X\beta) + \lambda \|\beta\|_2^2 \propto \min_{v} L(XX^T v) + \lambda v^T X X^T v$$

- It follows that:

$$\min_v L(XX^Tv) + \lambda \left\| X^Tv \right\|_2^2 = \min_v L(XX^Tv) + \lambda (X^Tv)^T (X^Tv)$$

$$= \min_v L(XX^Tv) + \lambda v^T XX^T v$$

- The optimization problem depends only on XX^T (dot products between data points)!

- But we can generalize the XX^T (the dot products between data points)!

- Instead of XX^T, let's consider $\phi(X)\phi(X)^T$ where

$$\phi : \mathbb{R}^p \to \mathbb{R}^{HUGE}$$

- New optimization problem:

$$\min_v L(\phi(X)\phi(X)^T v) + \lambda v^T \phi(X)\phi(X)^T v$$

Example: Ridge Regression (p >> n)

- Optimization problem:

$$\min_{v} \left\| Y - \phi(X)\phi(X)^T v \right\|_2^2 + \lambda v^T \phi(X)\phi(X)^T v$$

- This implies:

$$\Rightarrow v^* = [K + \lambda I_n]^{-1} Y$$

where $K = \phi(X)\phi(X)^T$.

- Recall: $\hat{\beta}_{Ridge} = (\phi(X)^T\phi(X) + \lambda I)^{-1}\phi(X)^T Y$

- **Question**: how do we solve for $\phi(X)$?

- **Answer**: we don't need to compute $\phi(X)$! We only need the kernel (Gramm) matrix:

$$K := (k(x_i, x_j))_{i,j} = (\phi(x_i)^T \phi(x_j))_{i,j}$$

- Our optimization problem reduces to:

$$\min_v L(Kv) + \lambda v^T K v$$

What is a kernel (k)?

- A kernel function must be symmetric and positive semi-definite (Mercer's Theorem)
- They can be regarded as generalized dot products.
- Kernels are a measure of similarity between data points

Polynomial kernel: $k(x, y) = (x^T y + 1)^q$

Example: Take $p, q = 2$.

$$
\begin{aligned}
k(x, y) &= (x^T y + 1)^2 \\
&= (x_1 y_1 + x_2 y_2 + 1)^2 \\
&= (x_1 y_1 + x_2 y_2 + 1)(x_1 y_1 + x_2 y_2 + 1) \\
&= 1 + 2x_1 x_2 + 2x_2 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 x_2 y_2 \\
&= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1 x_2, x_2^2)(1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, \sqrt{2}y_1 y_2, y_2^2)^T \\
&= \phi(x)^T \phi(y)
\end{aligned}
$$

- Large p? Large q?

Radial Basis Kernel: $k(x, y) = e^{-\|x-y\|^2}$

$$k(x, y) = e^{-(x-y)^2} = e^{-x^2} e^{-y^2} \sum_{k=0}^{\infty} \frac{2^k x^k y^k}{k!}$$

- Infinite dimensional!
- Not an issue because we don't need to compute $\phi(X)$ $(n \times \infty)$. Only need to compute K $(n \times n)$!

Advantages:

- Generate predictions from a (very) high dimensional space without ever explicitly operating in that space
- Large computation gain when $p >> n$.

Disadvantages:

- K is an $(n \times n)$ matrix – significant memory and computation requirements when n is large
- The choice of kernel is not always obvious
- We often can't recover β^*

$$\beta^* = \phi(X)^T (K + \lambda I_n)^{-1} Y$$

THANK YOU!

- Bishop, Christopher M. "Pattern recognition." Machine Learning 128 (2006): 1-58.
- "Kernels for Classification and Regressions." https://people.eecs.berkeley.edu/ russell/classes/cs194/f11/lectures/CS194
- "The Kernel Trick." https://people.eecs.berkeley.edu/ jordan/courses/281B-spring04/lectures/lec3.pdf