Deep Learning

Deeper, Better, _____ , Stronger than Machine Learning

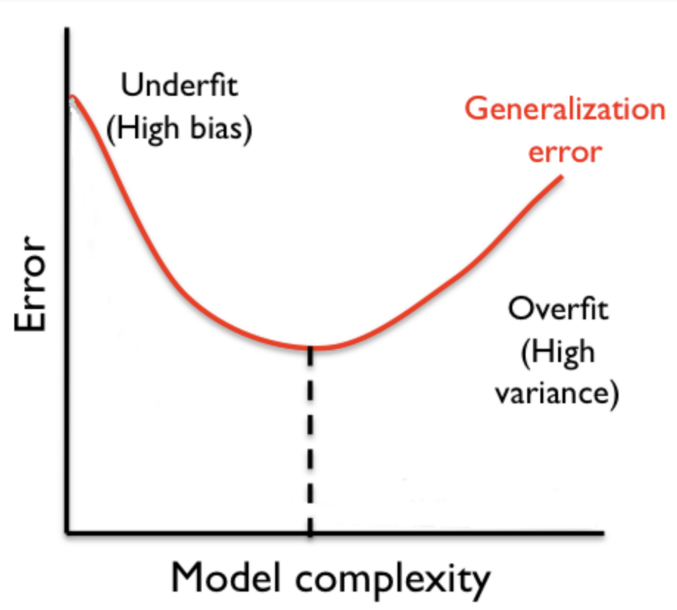
Valentin Barriere

Universidad de Chile – DCC

CC6204, Primavera 2024

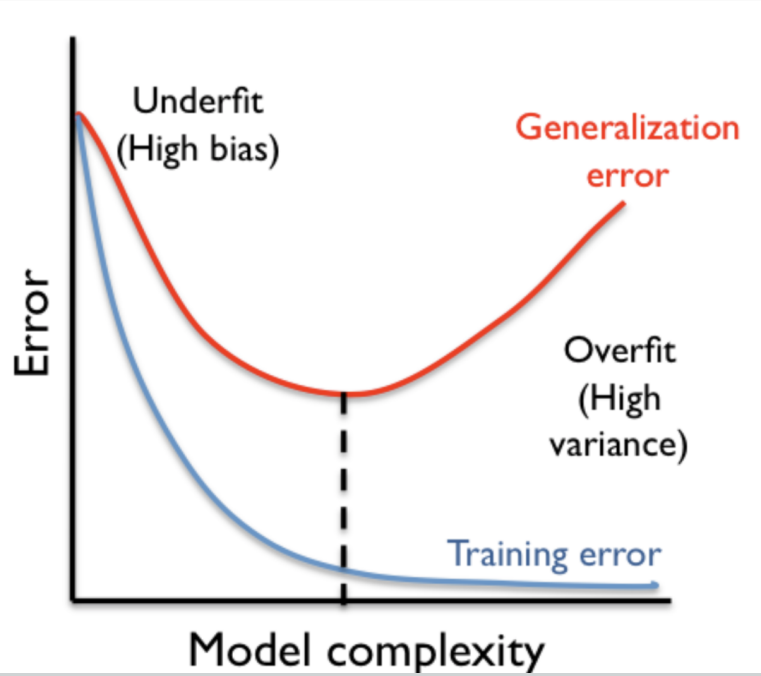
Regularization

Overfitting and Underfitting



- Depending on the model's complexity (e.g., training time), a different behavior is observed.
- Less complex models are learned easily, but the approximation error can be large (underfitting).
- Very complex models may have the correct objective but a large estimation error (overfitting).

Overfitting: Problem

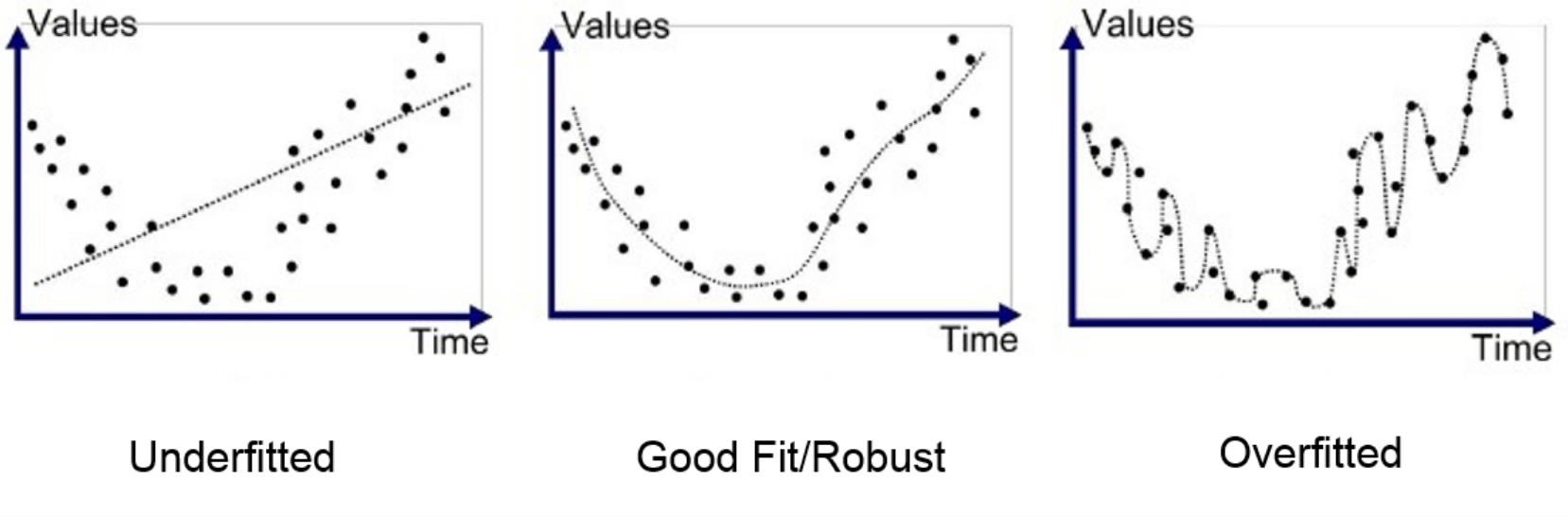


Error and Risks

- The empirical risk (error on the training set) decreases as the model complexity increases.
- The real risk (error on observations from a new set) is very different. We have a generalization problem!

- Overfitting: the learned parameters are too specific to the training set.
- A criterion different from the training set error should be used.

Overfitting: Complexity



Complexity

- If the model is too simple, it no longer fits the data.
- If the model is too complex, it learns all the irregularities of the dataset $\mathcal{D}_n$: the objective is not to memorize the training data but to generalize to new data.
- Example: if the model is the middle curve plus a noise component not considered in the variables, the model on the right learns that noise.

Overfitting: Regularization

A solution to combat the non-generalization problem: regularization

Principle

- The empirical risk of an estimator selected from a family of functions given the data is biased.
- By adding a penalty related to the model complexity f_θ, we can reduce it and minimize overfitting:

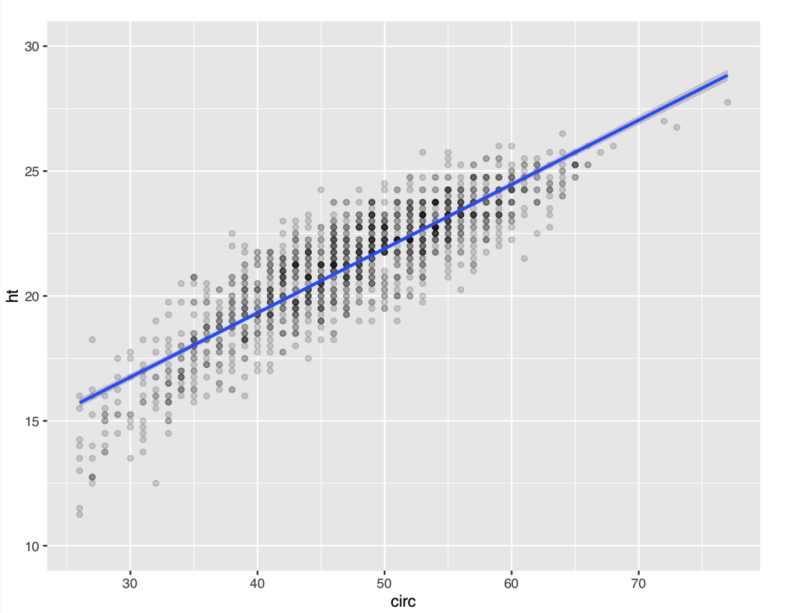
$$\mathcal{R}_n(f_\theta) \rightarrow \mathcal{R}_n(f_\theta) + pen(\theta)$$

- Then, in the risk:

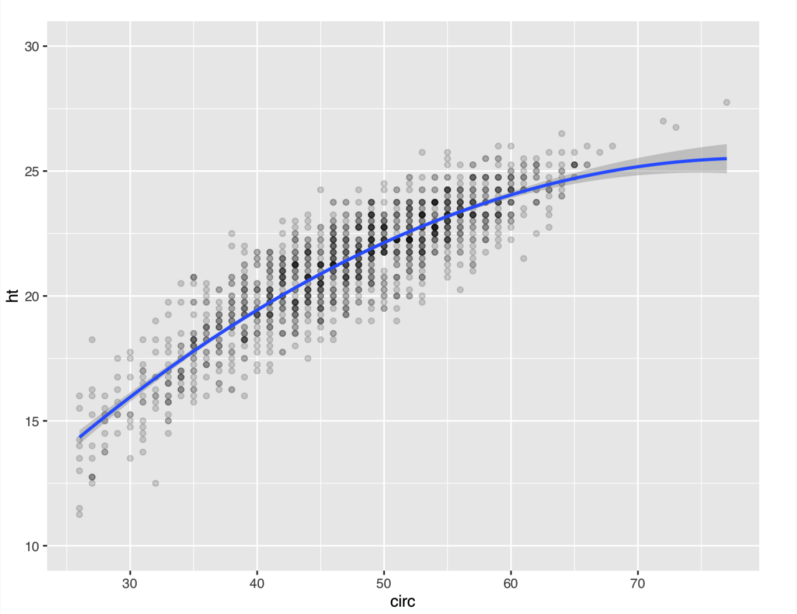
$$\arg \min_{f_\theta, \theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(\mathbf{X}_i)) + pen(\theta)$$

Allows using a large model with less overfitting.

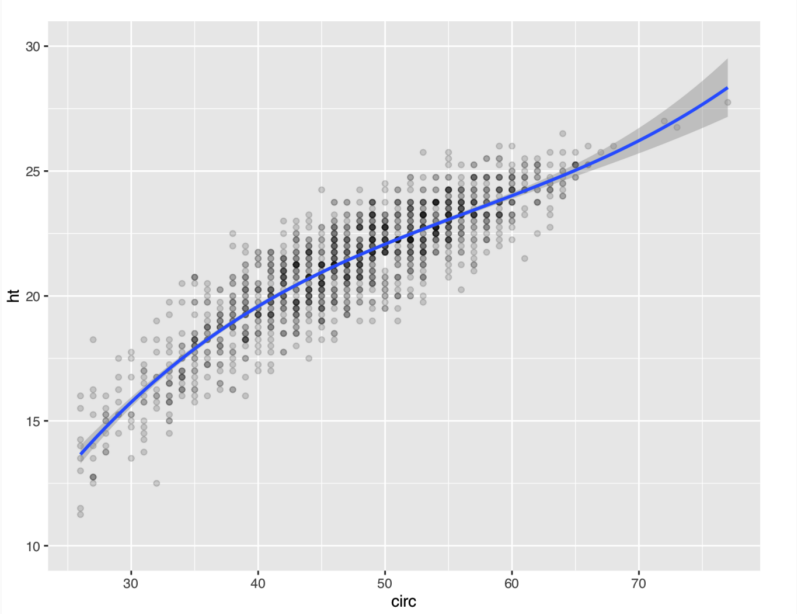
Overfitting: Regularization



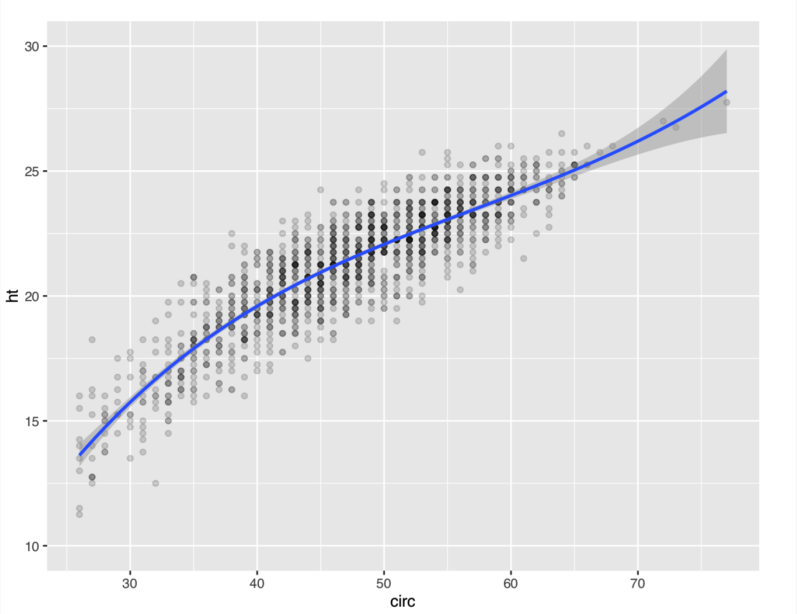
Overfitting: Regularization



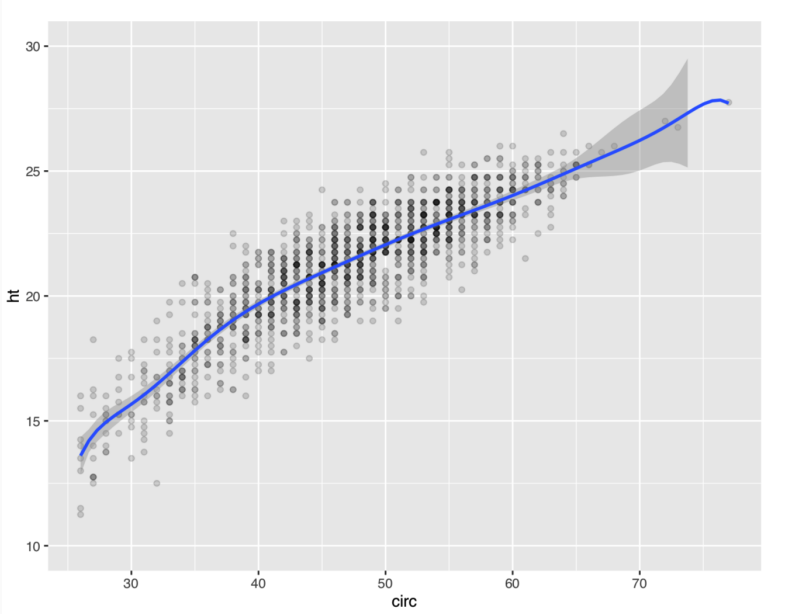
Overfitting: Regularization



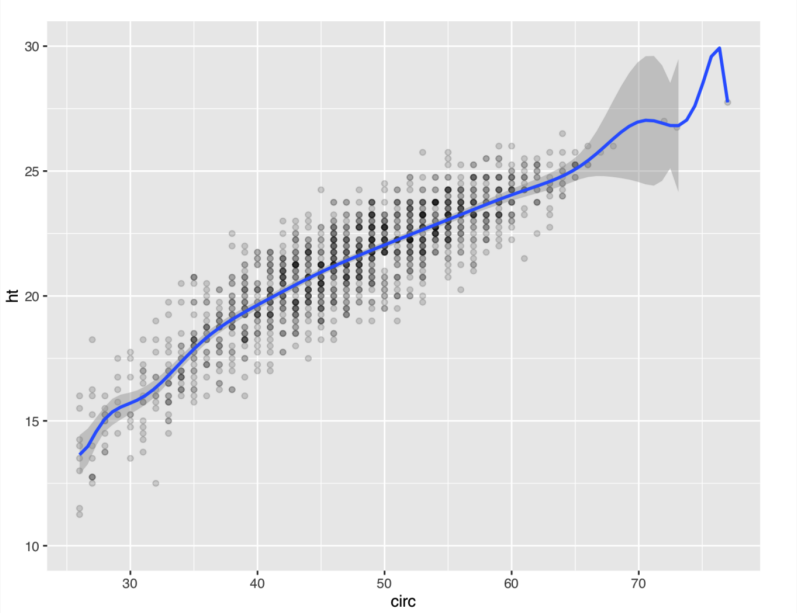
Overfitting: Regularization



Overfitting: Regularization



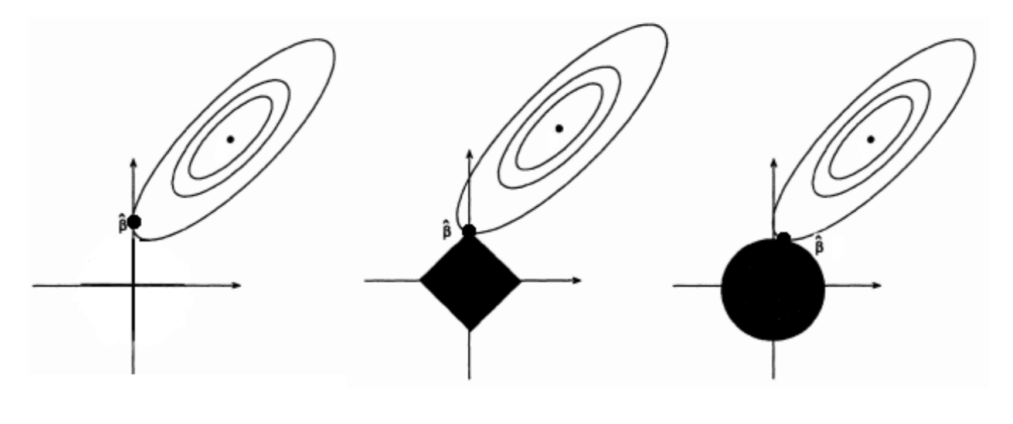
Overfitting: Regularization



Regularization

- Principle of parsimony (Occam's razor): the simpler and functional a model is, the better.
- It helps ignore too many specificities of the data.
- Intuition: reduce the norm of the model or the number of coefficients, number of branches in the graph (pruning).

Overfitting: Regularization



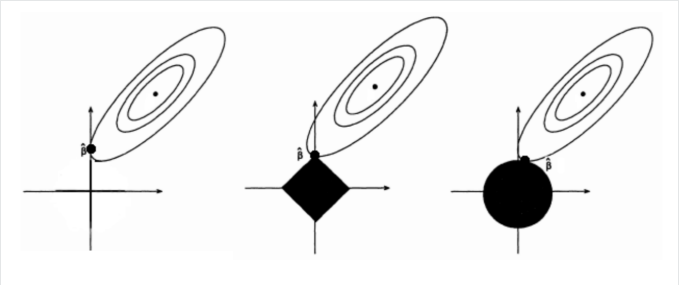
Classic Regularizations

- AIC: $pen(\theta) = \lambda||\theta||_0$ (non-convex, parsimonious, rarely used)
- Ridge: $pen(\theta) = \lambda||\theta||_2^2$ (convex, non-parsimonious, ℓ_2)
- Lasso: $pen(\theta) = \lambda||\theta||_1$ (convex, parsimonious, ℓ_1)
- Elastic Net: $pen(\theta) = \lambda_1||\theta||_1 + \lambda_2||\theta||_2$ (convex, parsimonious)

- Simple optimization if the cost (regularization) is convex.
- **Need to specify the λ values**, which become new hyperparameters.

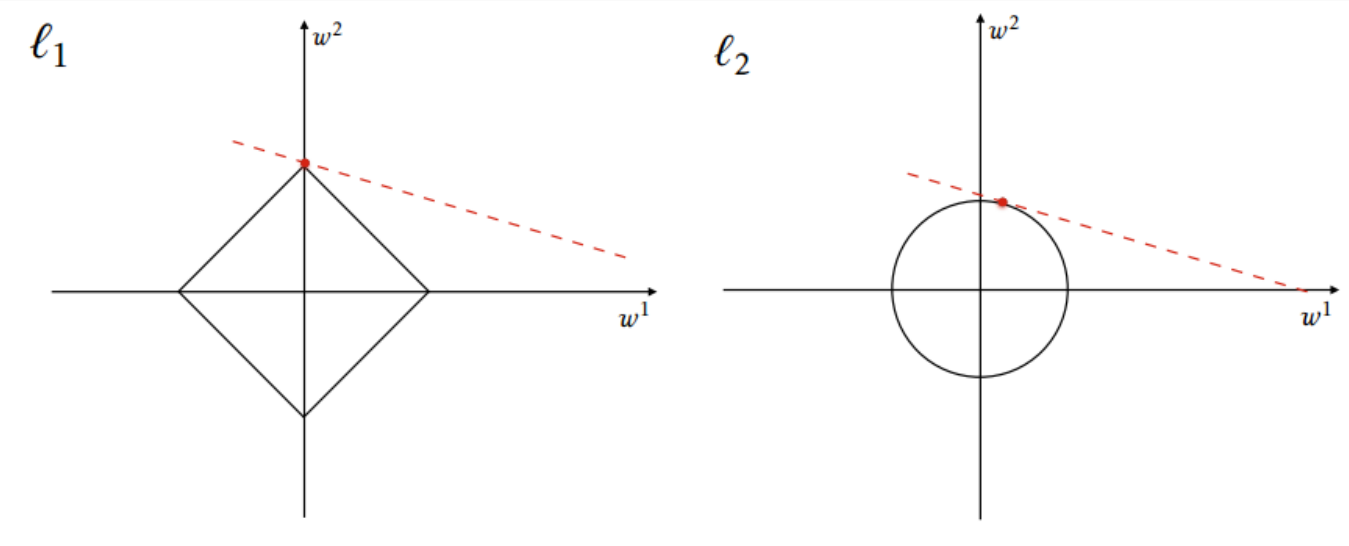
Parsimony Intuition

Lasso induces parsimony

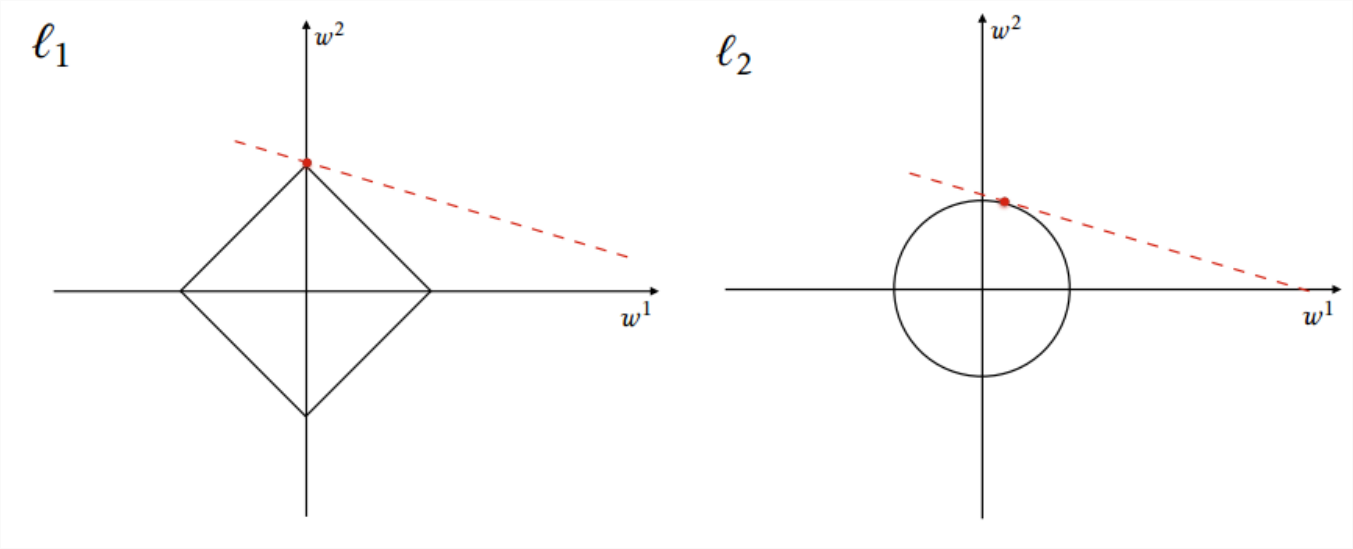


In black $\mathcal{B}^n = \{\mathbf{x}/\mathbf{x} \in \mathbb{R}^d, \text{ and } \|\mathbf{x}\|_n < 1\}$ for $n = 0, 1, 2$ in $\mathbb{R}^2$

- In high dimensions, most of $\mathcal{B}^1$ concentrates on the axes. This is equivalent to having zero values for other axes.



Parsimony Intuition



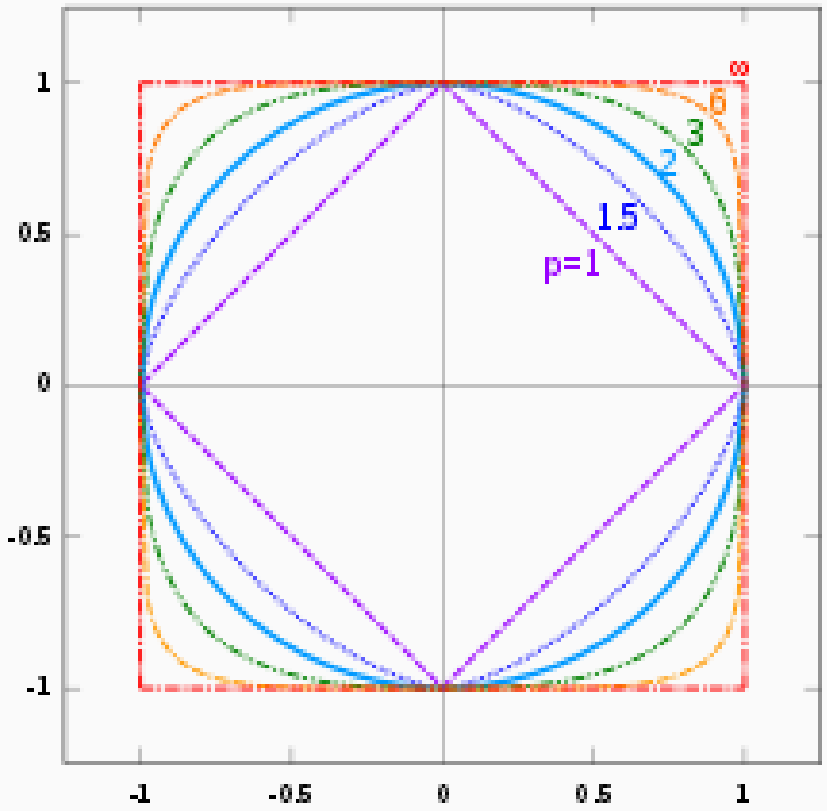
- The solution to the problem lies on the red line: any point on it is acceptable.
- If we have a regularization, we want the solution that gives us a minimum value of the norm.
- Intuition: the ℓ_1 norm will provide solutions on the axes, which in this case means that parameter w^1 equals zero!

Parsimony Intuition: Norms

Exercise: We have seen the 1-ball and the 2-ball. What shape does the infinite ball have?

Parsimony Intuition: Norms

Exercise: We have seen the 1-ball and the 2-ball. What shape does the infinite ball have?



$\|\mathbf{x}\|_n = \left(\sum_{i=1}^{d} |x_i|^n\right)^{\frac{1}{n}}$, hence $\|\mathbf{x}\|_\infty = \max_i |x_i|$

Overfitting in Neural Nets

As we said, we do not want the model to learn the specifics details of our training dataset.

Strategies to counter overfitting:

- Increase the amount of data to add variability and complicate the memorization process
- Use simpler models (smaller)

Overfitting in Neural Nets

As we said, we do not want the model to learn the specifics details of our training dataset.

Strategies to counter overfitting:

- Increase the amount of data to add variability and complicate the memorization process **But... data is costly!**
- Use simpler models (smaller)

Overfitting in Neural Nets

As we said, we do not want the model to learn the specifics details of our training dataset.

Strategies to counter overfitting:

- Increase the amount of data to add variability and complicate the memorization process
- Use simpler models (smaller) **But... we want large networks to solve complex tasks!**

ℓ_2 regularization

For a loss function $\mathcal{L}$, the classical way to regularize in Deep Learning is to use an ℓ_2 regularization:

$$\mathcal{L}_{reg} = \mathcal{L} + \frac{\lambda}{2}||\theta||_2^2 = \mathcal{L} + \frac{\lambda}{2}\sum_i \theta_i^2$$

Also called Weight Decay

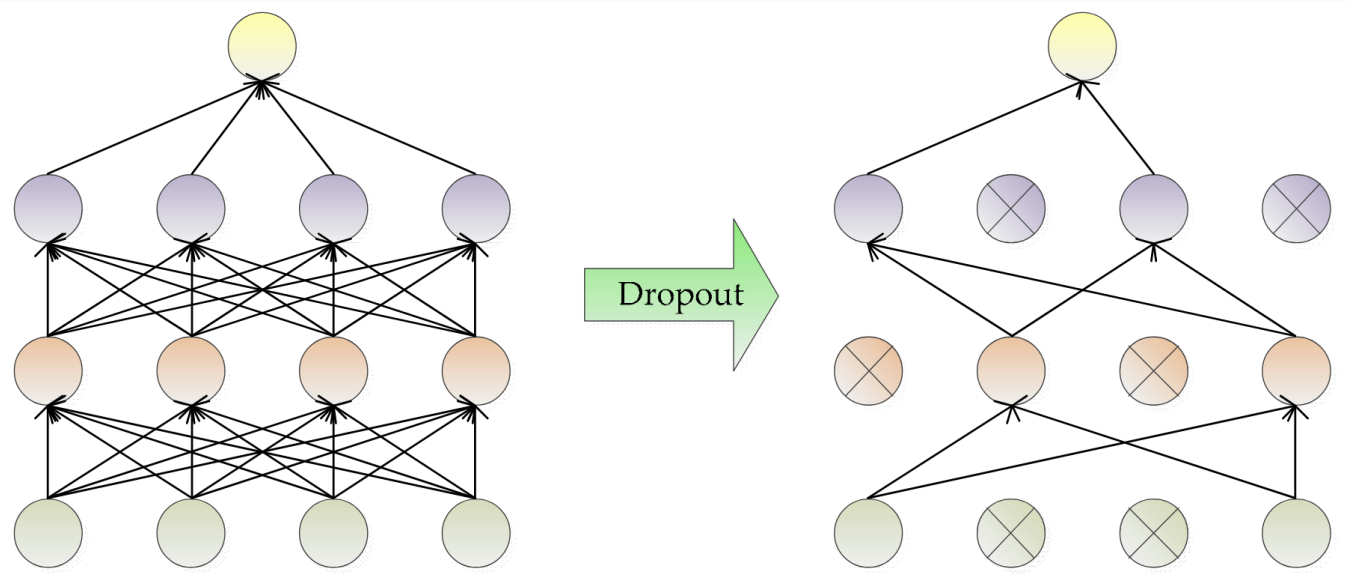
$$\theta_i = \theta_i - \alpha\left(\frac{\partial \mathcal{L}}{\partial \theta_i} + \lambda\theta_i\right)$$

$$\theta_i = \theta_i\left(1 - \alpha\lambda\right) - \alpha\frac{\partial \mathcal{L}}{\partial \theta_i}$$

Dropout: principle

What is Dropout?

- Dropout is a regularization technique used in neural networks to prevent overfitting, which was introduced by Geoffrey Hinton and his team in 2012.
- The idea is to randomly "drop out" (set to zero) a fraction of neurons **during the training process**.



At each training iteration, a random subset of neurons is ignored.

Dropout: principle

Why Use Dropout?

- It prevents the model from becoming too dependent on specific neurons, forcing the network to learn more robust features, and improves generalization to unseen data.
- At each iteration, the network effectively trains a different subset of the model, creating an ensemble effect

For patients with depression, these paths might represent negative thoughts or emotions that keep getting triggered.

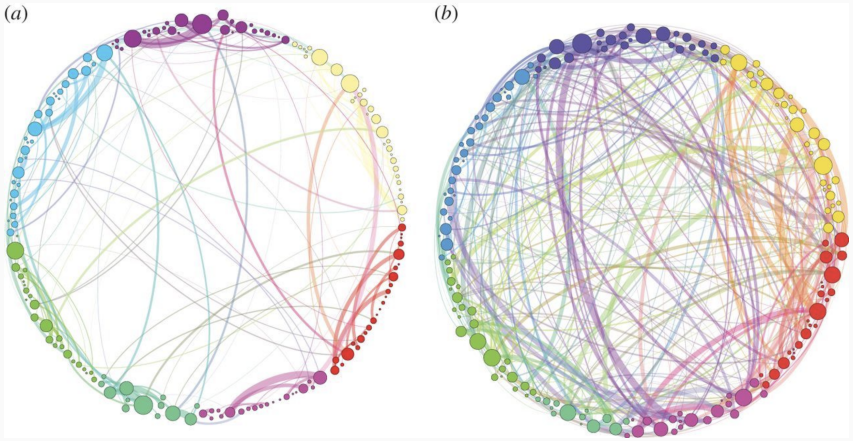


Figure 1: Visulization from Petri et al. (2014)

Data augmentation

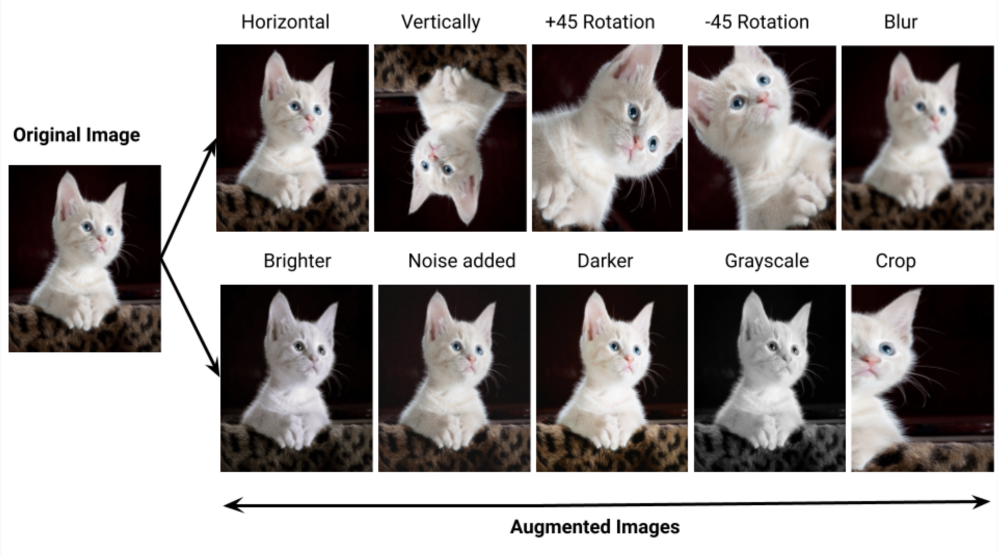
Another solution is to simply use more data! But it can get costly to obtain, so you can artificially create some using data augmentation.

- More data always reduce overfitting, because we introduce variability
- A priori knowledge needs to be used in order to introduce meaningful data, otherwise you perturbate the data distribution
  - A cat seen in the mirror is still a cat!
  - North hemisphere satellite data always faces the sun the same way

Data augmentation

Another solution is to simply use more data! But it can get costly to obtain, so you can artificially create some using data augmentation.

- More data always reduce overfitting, because we introduce variability
- A priori knowledge needs to be used in order to introduce meaningful data, otherwise you perturbate the data distribution
  - A cat seen in the mirror is still a cat!
  - North hemisphere satellite data always faces the sun the same way
- In the case of **images**, typical transformation are: Small rotations, Flipping, Random crops, Color Jittering, Noise addition, ...



Data augmentation

Another solution is to simply use more data! But it can get costly to obtain, so you can artificially create some using data augmentation.

- More data always reduce overfitting, because we introduce variability
- A priori knowledge needs to be used in order to introduce meaningful data, otherwise you perturbate the data distribution
  - A cat seen in the mirror is still a cat!
  - North hemisphere satellite data always faces the sun the same way
- In the case of **images**, typical transformation are: Small rotations, Flipping, Random crops, Color Jittering, Noise addition, ...
- In the case of **texts**: Gender swap, Random punctuation insertion, Lexicon-based perturbation (synonym/entities replacement), Back Translation, ...
- In the case of **audio**: Time Shifting, Pitch shifting, Reverberation, Amplitude scaling, Noise addition, ...

Audio examples: original file, with reverb, or with reverb and noise.

Questions?

References i