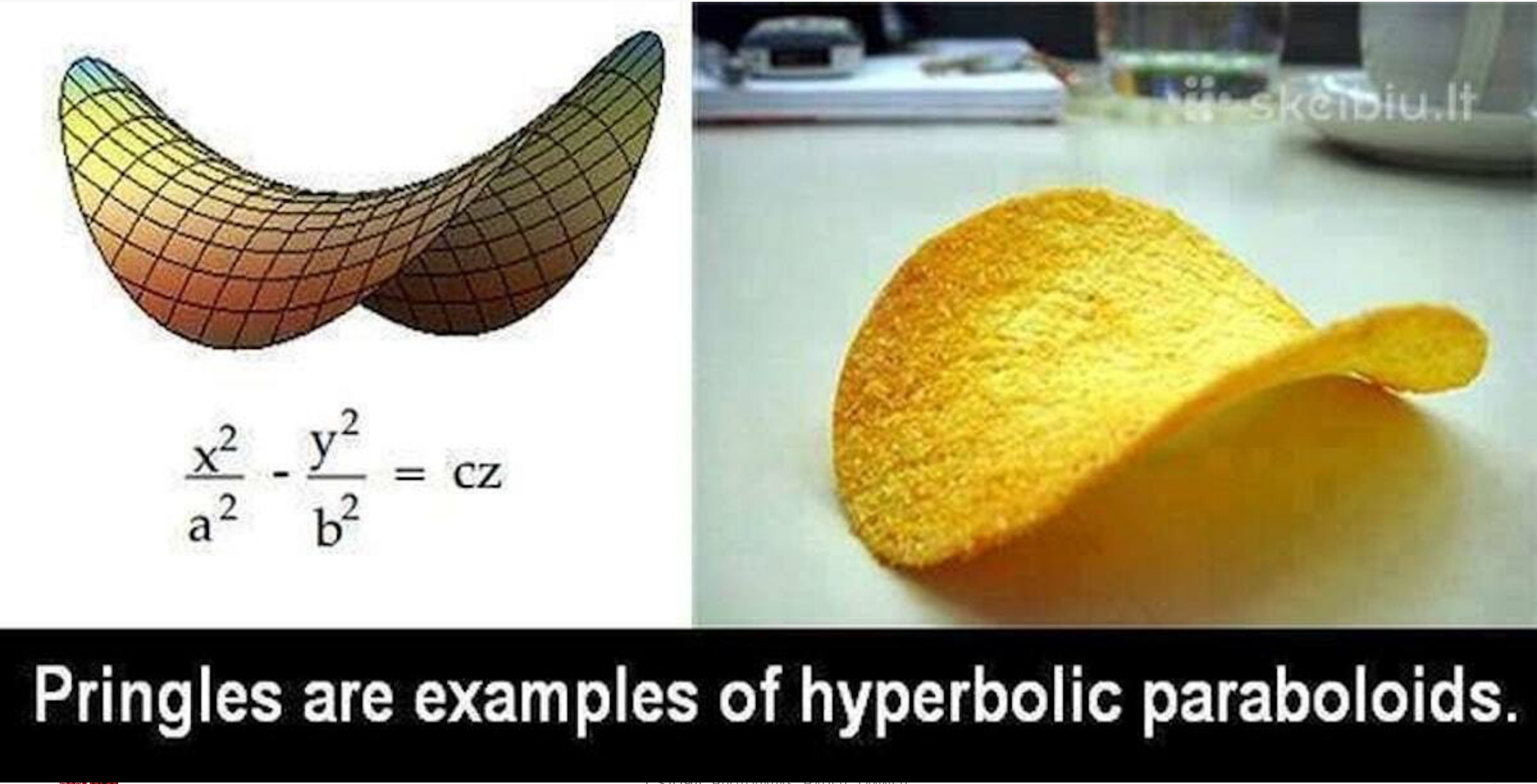
# Neural Network Regularization

## CS109A Introduction to Data Science

Pavlos Protopapas, Kevin Rader and Chris Tanner


$\frac{x^2}{a^2} - \frac{y^2}{b^2} = cz$
skelbiu.lt
Pringles are examples of hyperbolic paraboloids.

# Outline

**Regularization of NN**

- Norm Penalties
- Early Stopping
- Data Augmentation
- Dropout

# Regularization

Regularization is any modification we make to a learning algorithm that is intended to **reduce its generalization error** but not its training error.

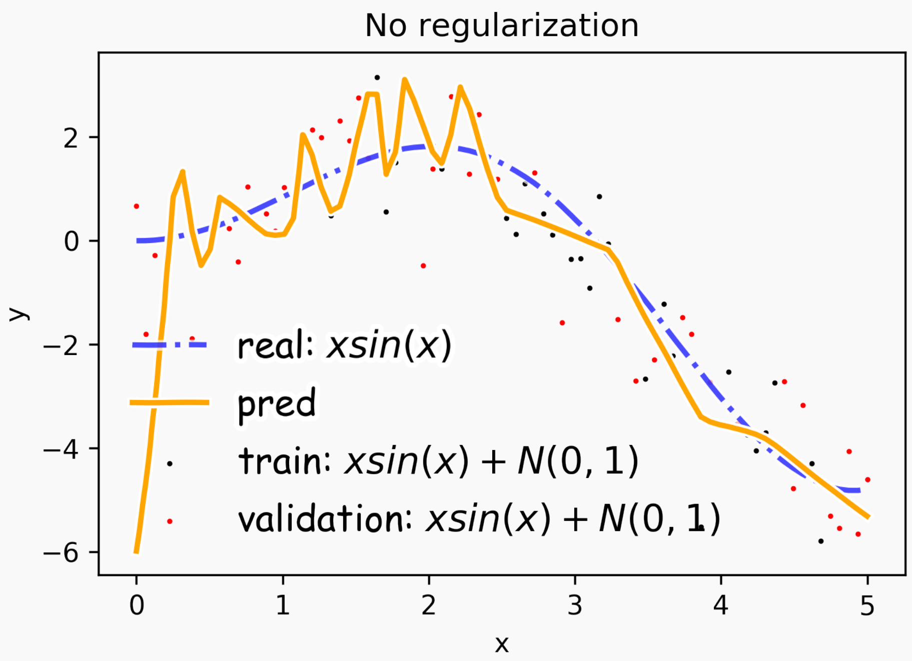
# Outline

**Regularization of NN**

- **Norm Penalties**
- Early Stopping
- Data Augmentation
- Dropout

# Overfitting

Fitting a deep neural network with 5 layers and 100 neurons per layer can lead to a very good prediction on the training set but poor prediction on validations set.



# Norm Penalties

We used to optimize:

$$L(W; X, y)$$

Change to …

$$L_R(W; X, y) = L(W; X, y) + \alpha\Omega(W)$$

Biases not penalized

$L_2$ regularization:

- Weights decay
- MAP estimation with Gaussian prior

$$\Omega(W) = \frac{1}{2} \| W \|_2^2$$

$L_1$ regularization:

- encourages sparsity
- MAP estimation with Laplacian prior

$$\Omega(W) = \frac{1}{2} \| W \|_1$$

# Norm Penalties

We used to optimize:

$$W^{(i+1)} = W^{(i)} - \lambda \frac{\partial L}{\partial W}$$

Change to:

$$L_R(W; X, y) = L(W; X, y) + \frac{1}{2}\alpha \parallel W \parallel_2^2$$

$$W^{(i+1)} = W^{(i)} - \lambda \frac{\partial L}{\partial W} - \lambda\alpha W^{(i)}$$

Weights decay in proportion to size

Biases not penalized

$L_2$ regularization:

- **Decay of weights**
- MAP estimation with Gaussian prior

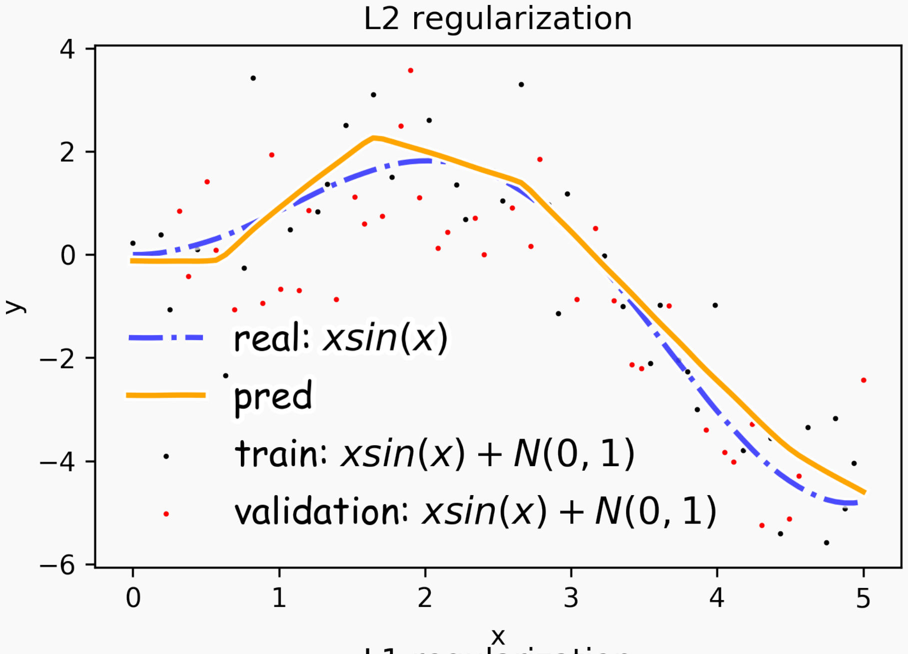
$$\Omega(W) = \frac{1}{2} \parallel W \parallel_2^2$$

$L_1$ regularization:

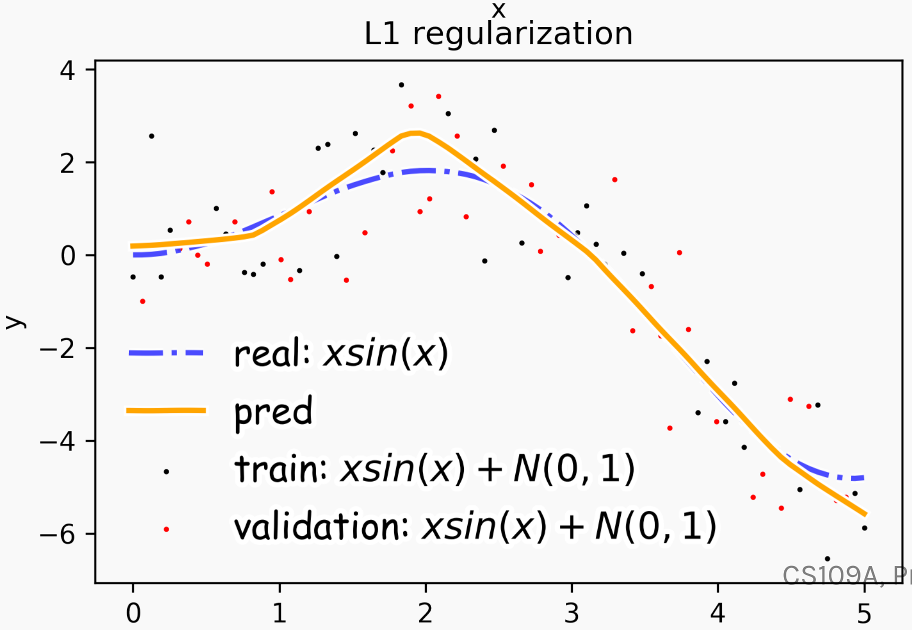
- encourages sparsity
- MAP estimation with Laplacian prior

$$\Omega(W) = \frac{1}{2} \parallel W \parallel_1$$

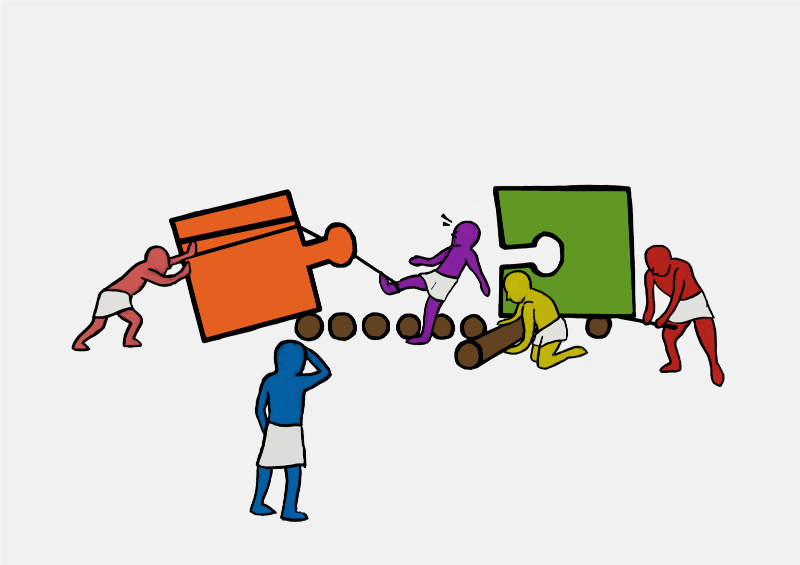
# Norm Penalties




$$\Omega(W) = \frac{1}{2} \parallel W \parallel_2^2$$




$$\Omega(W) = \frac{1}{2} \parallel W \parallel_1$$



# Exercise: Regularization using L1 and L2 Norm

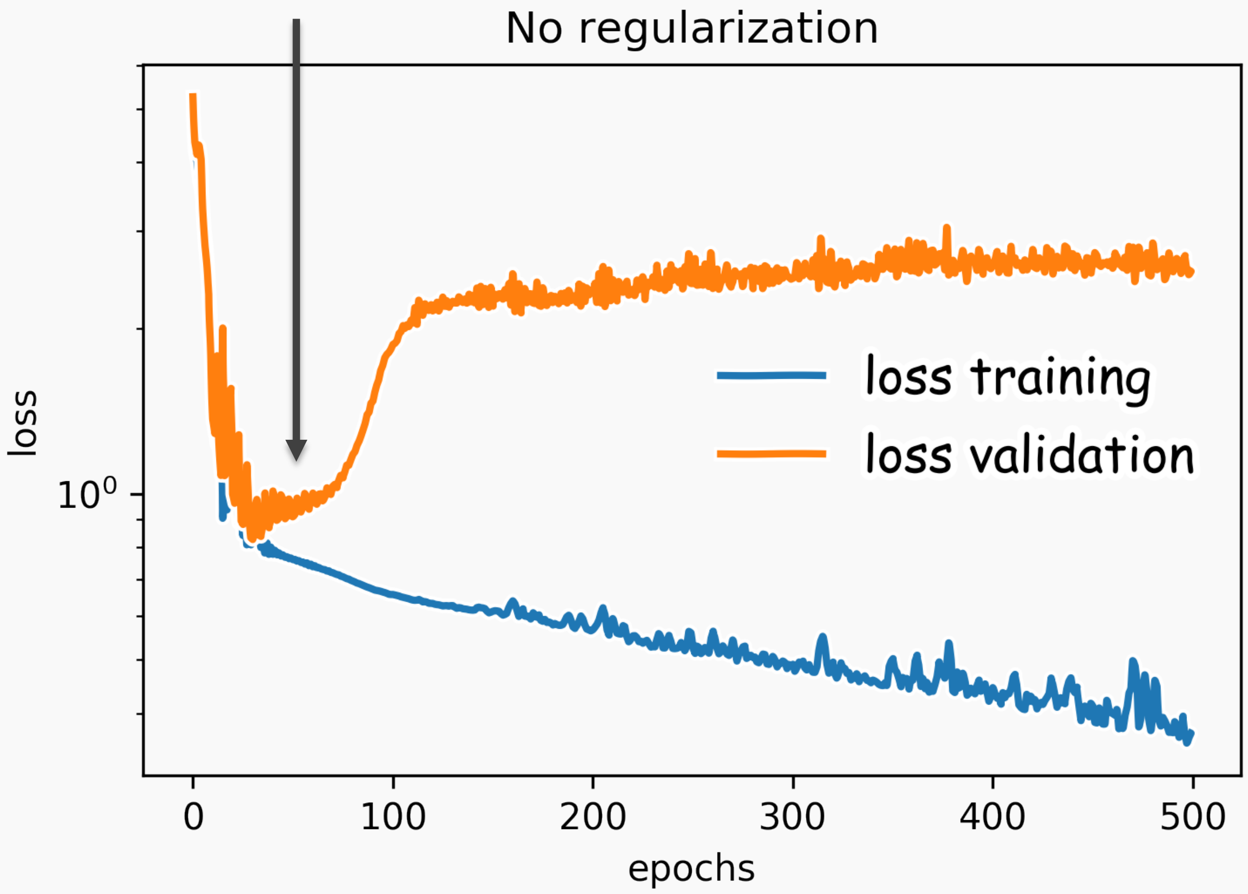
# Outline

**Regularization of NN**

- Norm Penalties
- **Early Stopping**
- Data Augmentation
- Dropout

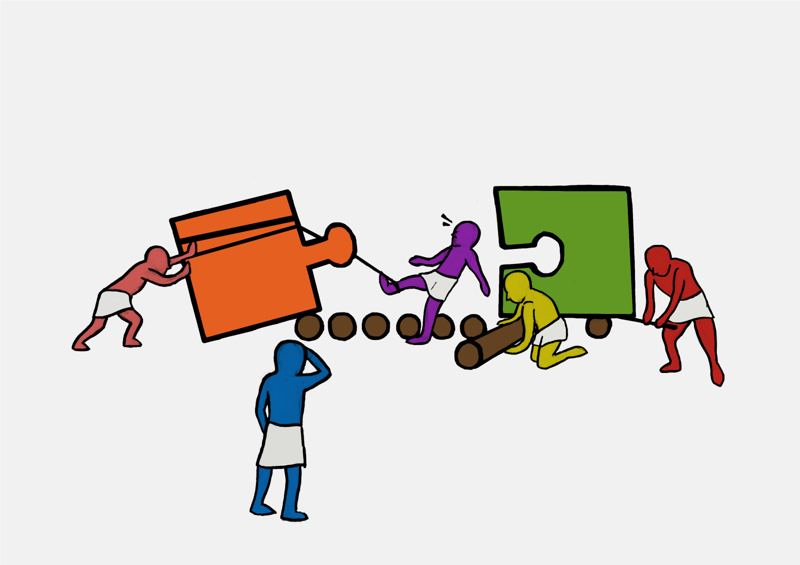
# Early Stopping

Early stopping: terminate while validation set performance is better. Sometimes is worth waiting a little before stopping. This is called `patience`.




Patience is defined as the number of epochs to wait before early stop if no progress on the validation set.

The patience is often set somewhere between 10 and 100 (10 or 20 is more common), but it really depends on the dataset and network.



Exercise: Early Stopping