# Small data technique I

## Lecture 15

Changho Suh

October 1, 2021

# Random forests (RFs)

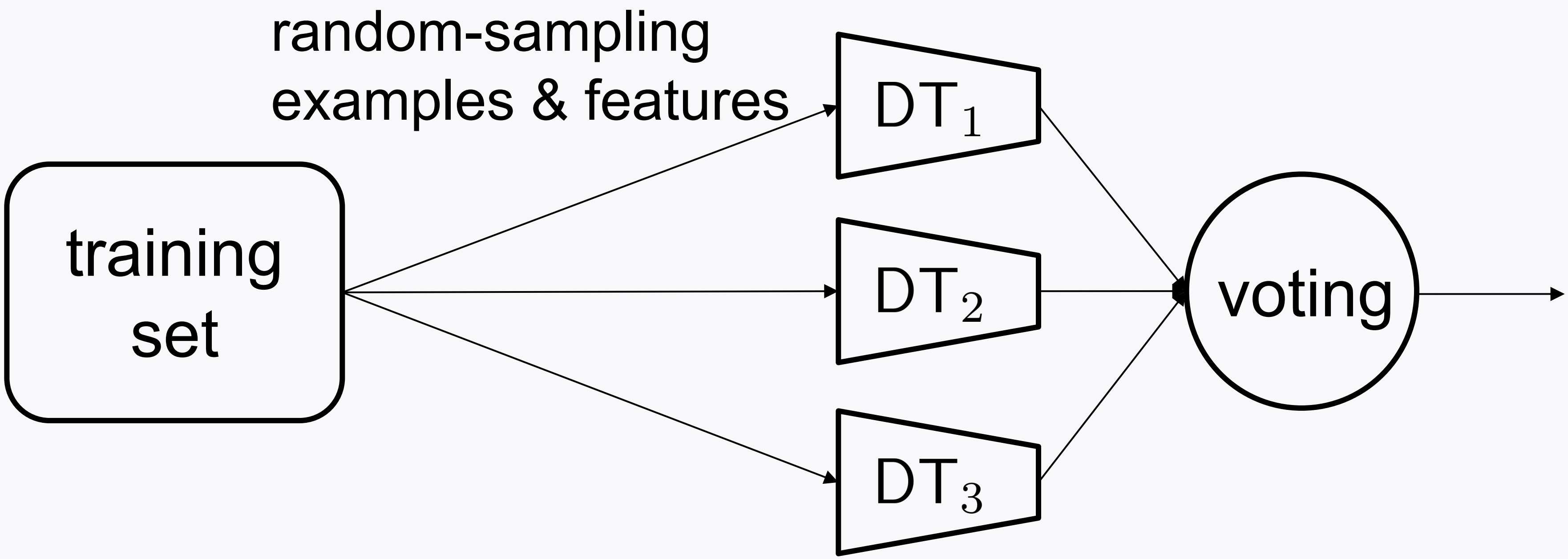
# Outline

1. Investigate **hyperparameters.**

2. Study a key measure for model *interpretation*:

   **Feature Importance**

# Hyperparameters




Two types:

**DT** hyperparameters + **additional** hyperparameters

# Hyperparameters

**DT** hyperparameters + **additional** hyperparameters

"max_depth"

"min_samples_split"

"min_samples_leaf"

"max_leaf_nodes"

"max_features"

"n_estimators"

# Default parameters

**DT** hyperparameters + **additional** hyperparameters

| | | | |
|---|---|---|---|
| “max_depth” | none | “max_features” | $\frac{\sqrt{\text{n\_features}}}{\text{n\_features}}$ |
| “min_samples_split” | 2 | “n_estimators” | 100 |
| “min_samples_leaf” | 1 | | |
| “max_leaf_nodes” | none | | |

# Hyperparameters vs. regularization

| **DT** hyperparameters | | + | **additional** hyperparameters | |
|---|---|---|---|---|
| "max_depth" | ↓ | | "max_features" | ↓ |
| "min_samples_split" | ↑ | | "n_estimators" | ↑ |
| "min_samples_leaf" | ↑ | | | |
| "max_leaf_nodes" | ↓ | | | |

→ More regularized.

# Hyperparameter search

Scikit-learn provides functions that ease search:

**`GridSearchCV`**

**`RandomizedSearchCV`**

Check details in PS.

# A measure for model interpretation

RFs have a **measure** that captures **the relative importance of each feature**:

**Feature Importance**

Can serve model interpretation.

# How to compute "feature importance"?

For each DT, first compute "node importance":

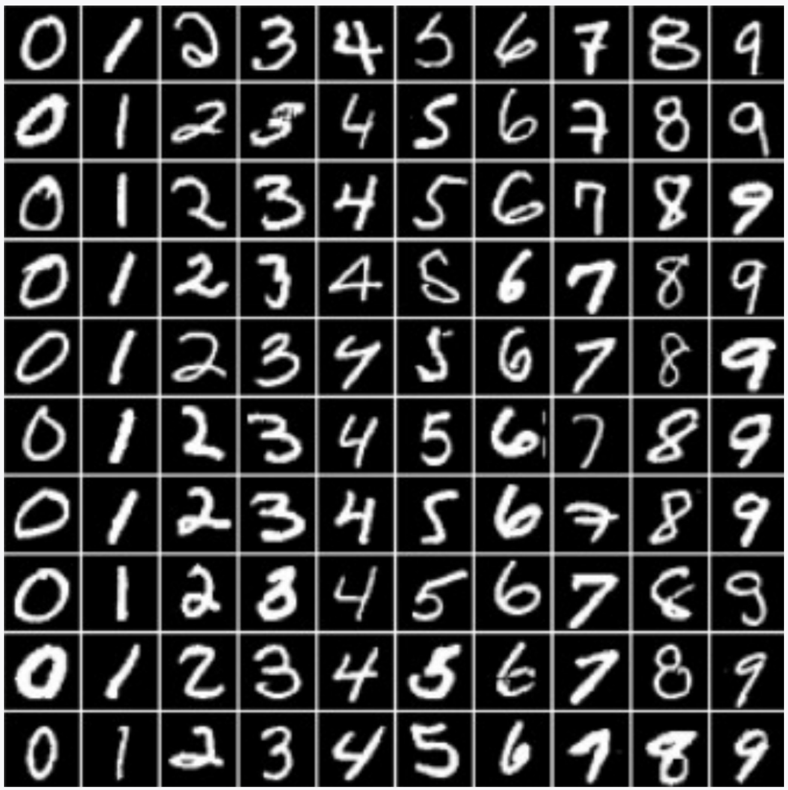
$$\mathsf{NI}_j = G_j - \frac{m_{j,\mathsf{left}}}{m_j} G_{j,\mathsf{left}} - \frac{m_{j,\mathsf{right}}}{m_j} G_{j,\mathsf{right}}$$

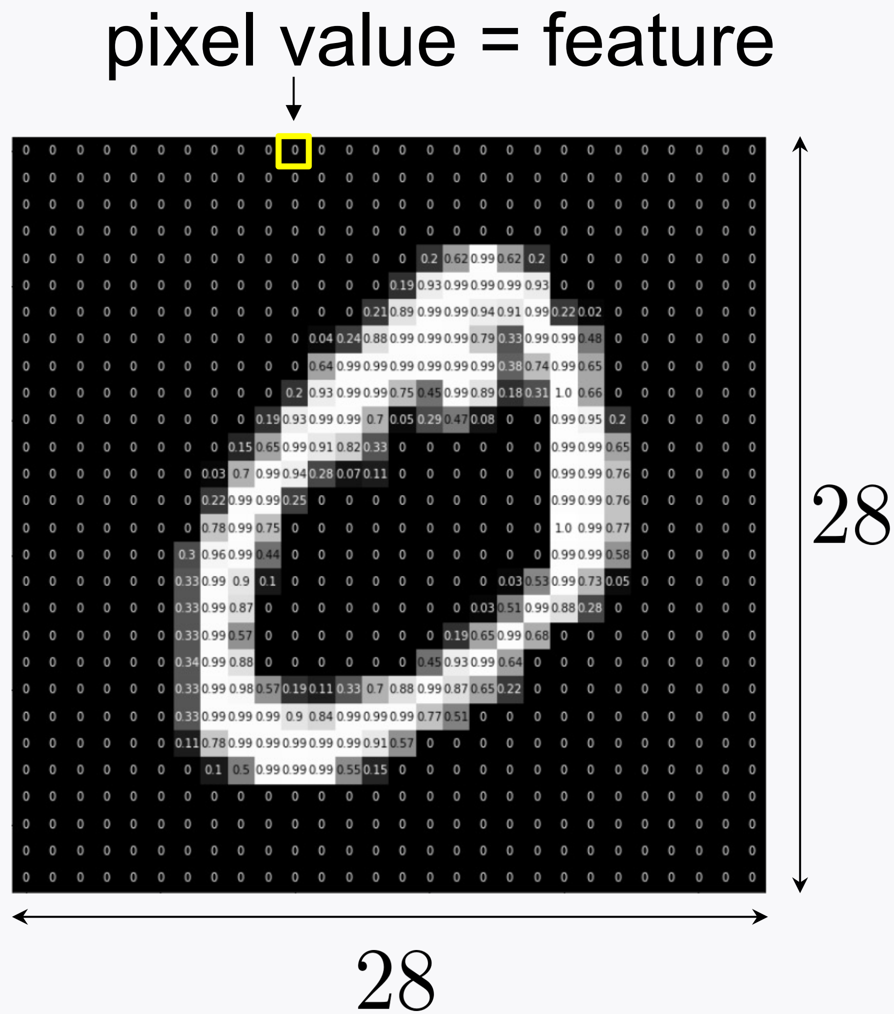
Then compute "feature importance" based on $\mathsf{NI}_j$:

$$\mathsf{FI}_k = \frac{\sum_{j:\text{w.r.t. } k} \mathsf{NI}_j}{\sum_j \mathsf{NI}_j}$$

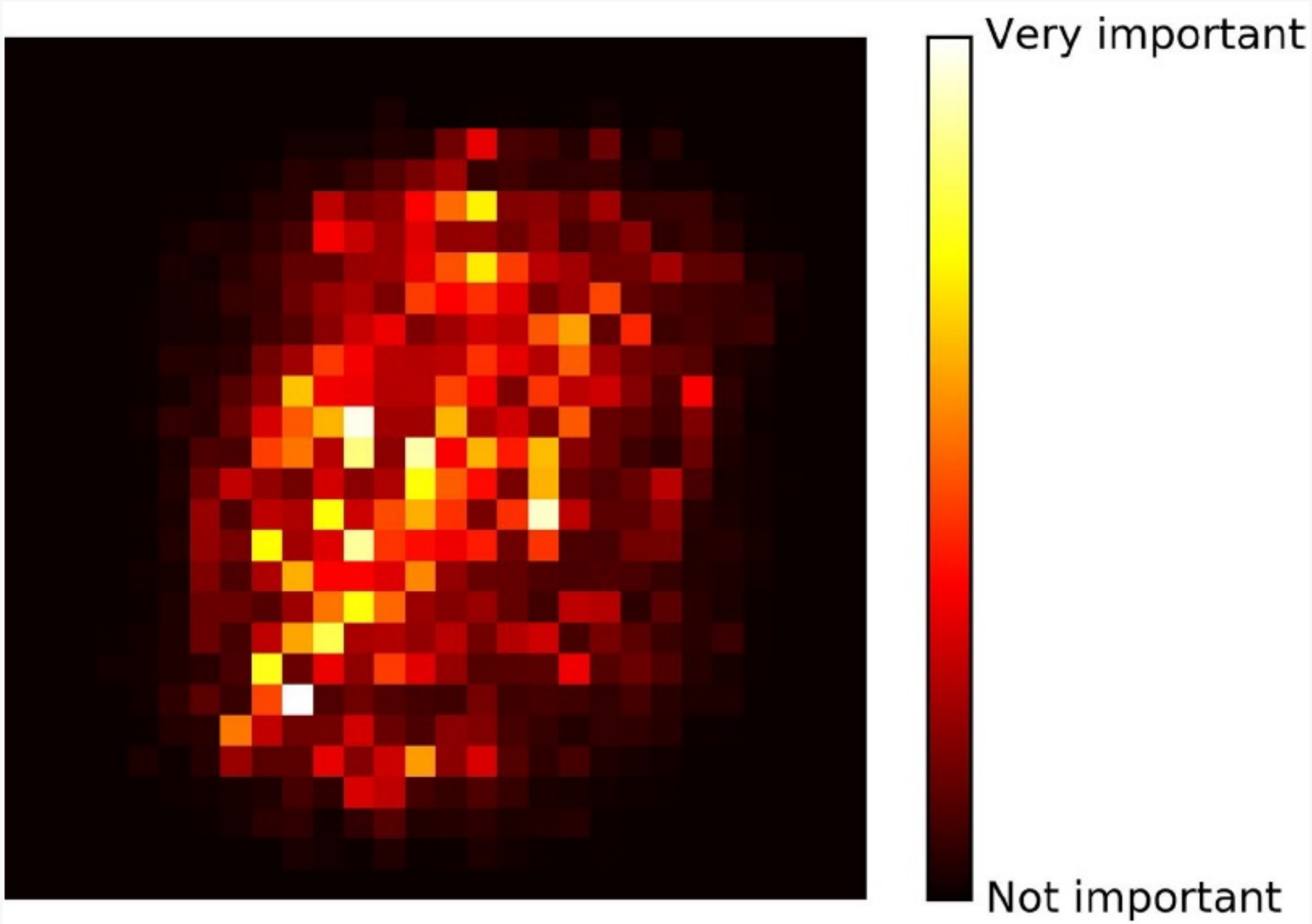
Average over all DTs.

# Example: MNIST

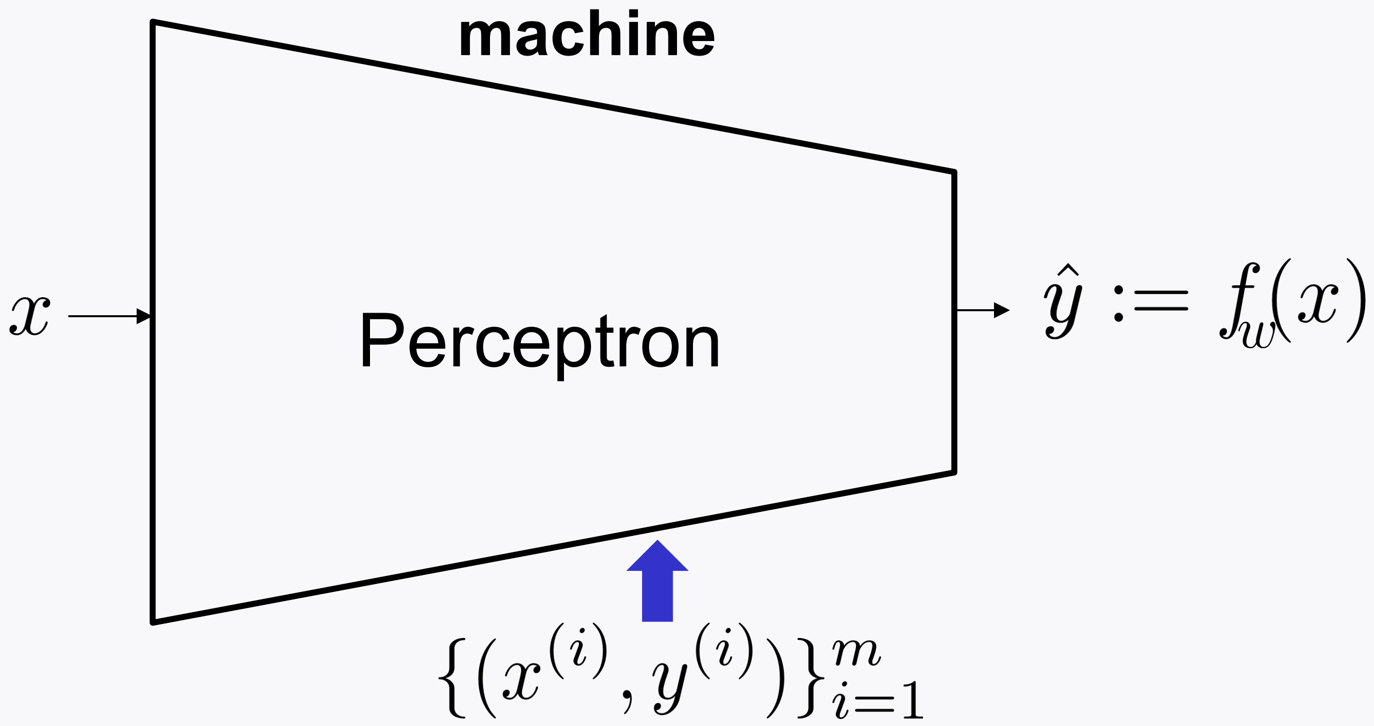




# MNIST pixel importance



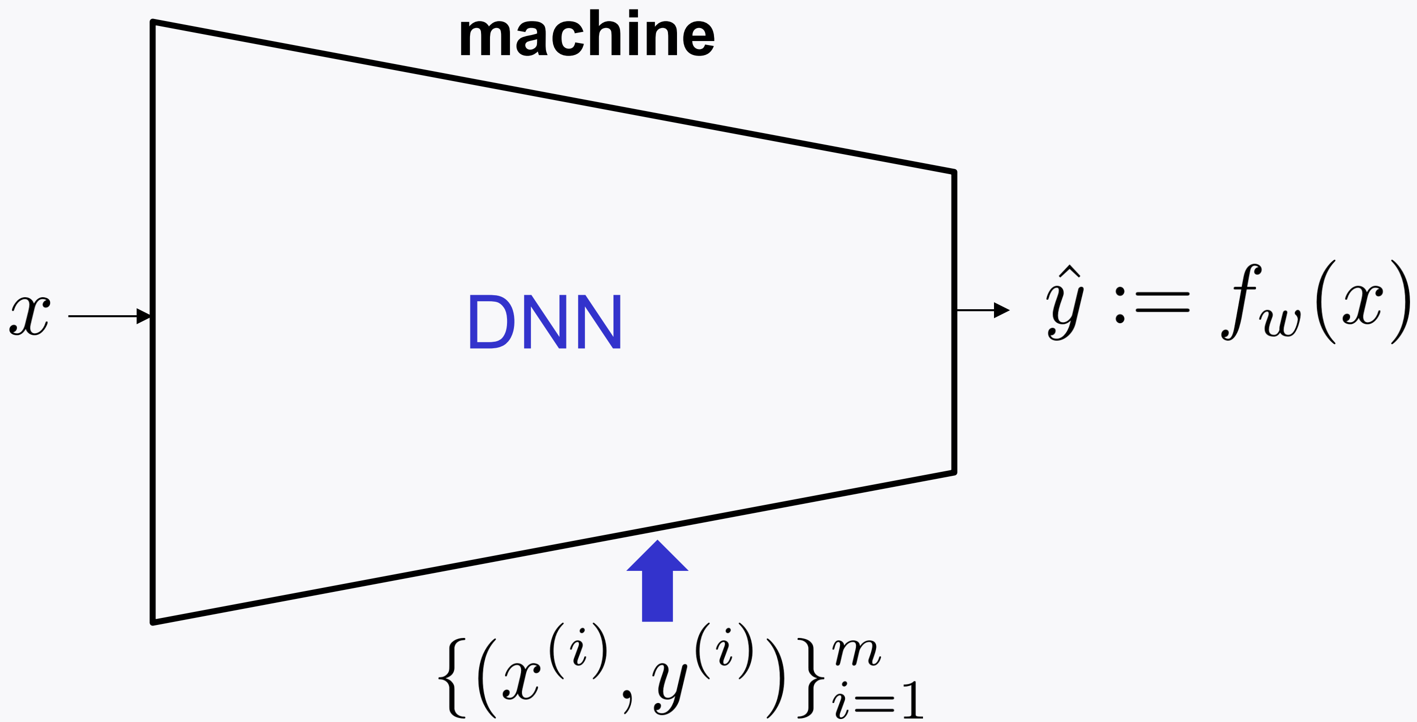
# Summary of Day 1 lectures




Linear activation + squared-error loss: **LS** classifier

Logistic acti. + cross entropy loss: **Logistic regression**

# Summary of Day 1 lectures




**Rule of thumb**: ReLU (@hidden); Logistic (@output)

Cross-entropy loss

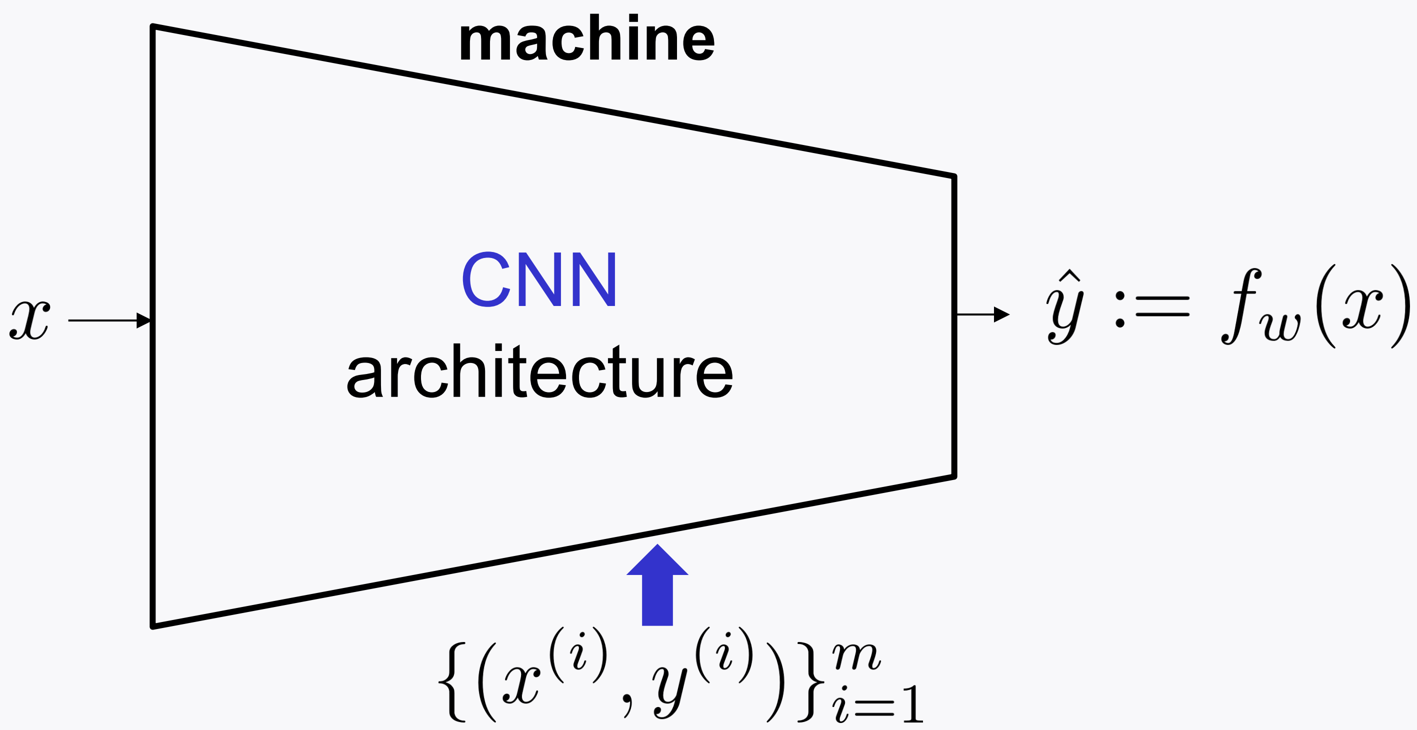
**Algorithm**: Gradient descent via backprop

# Summary of Day 2 lectures

Advanced techniques:

1. Data organization
2. Generalization techniques
3. Weight initialization
4. Techniques for training stability
5. Hyperparameter search
6. Cross validation

# Summary of Day 3 lectures

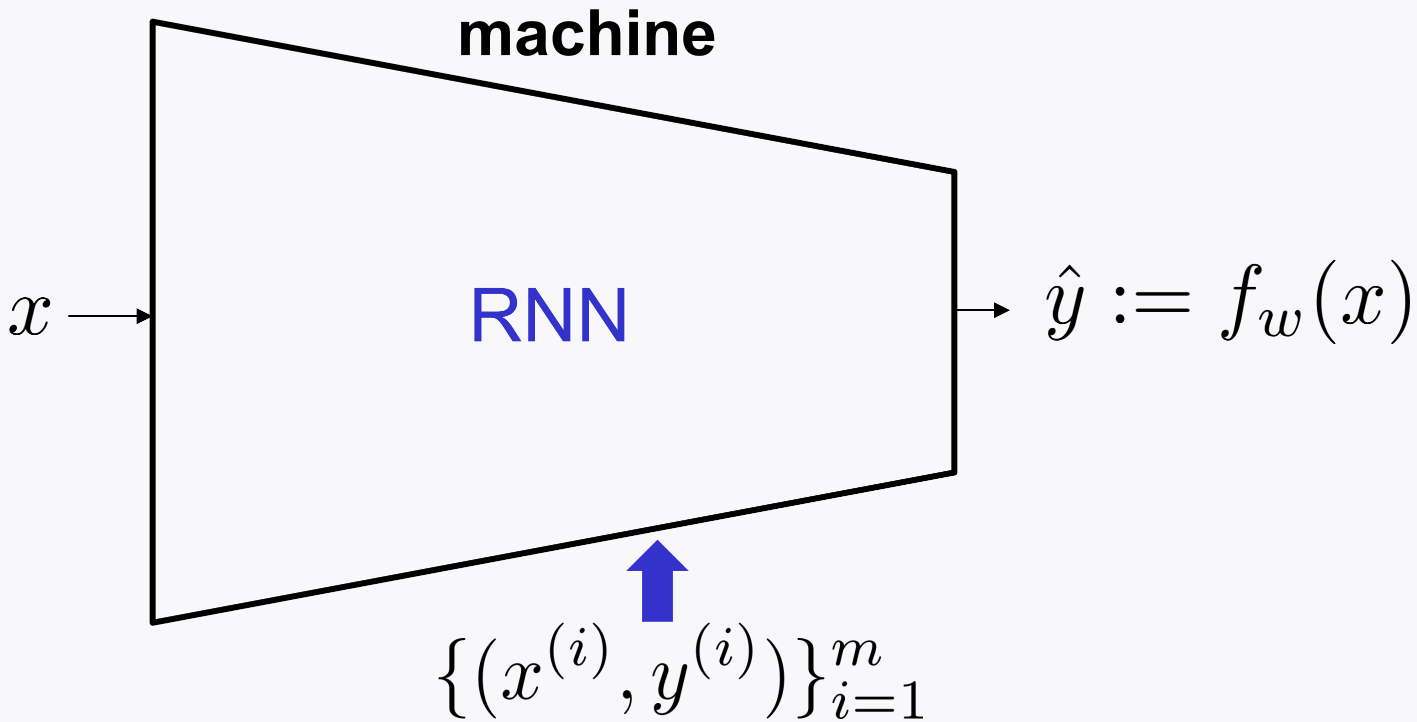



**Two key building blocks**: Conv layer & Pooling layer

**Design principles**: As a network is deeper,

1. Feature map sizes gets smaller.
2. # of feature maps gets bigger.

# Summary of Day 4 lectures



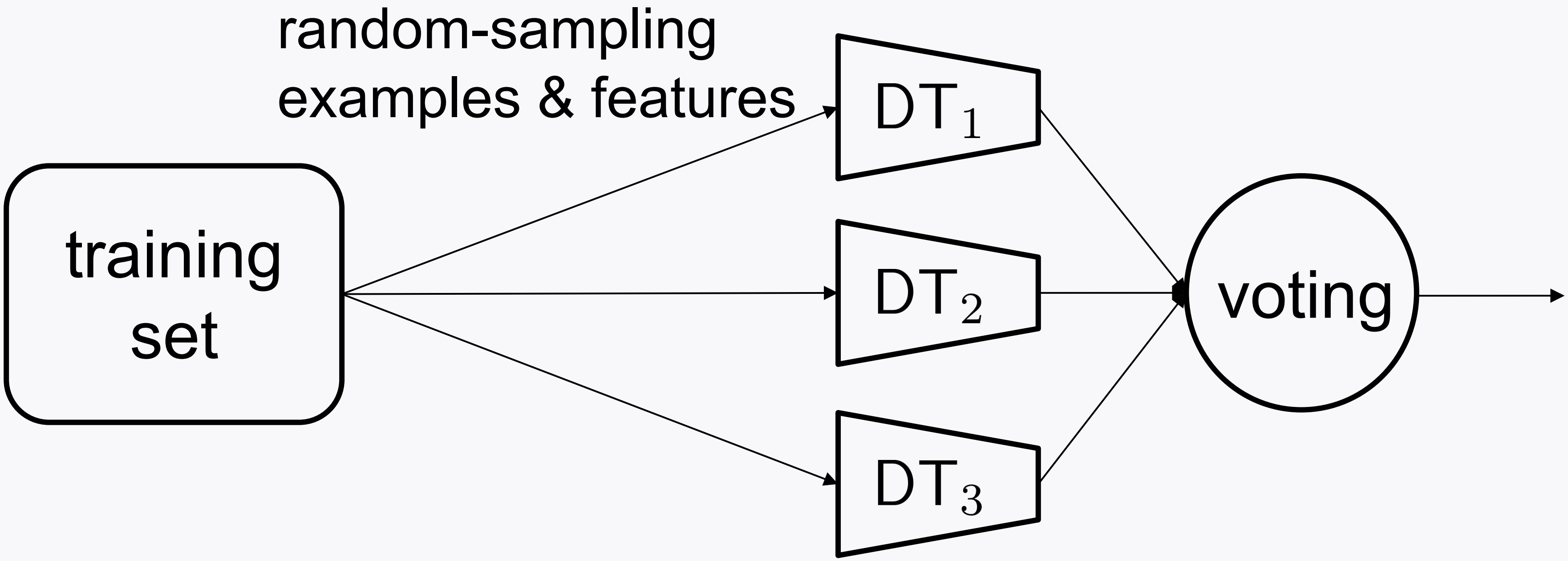

**Key building blocks**: Recurrent neurons & memory cell

**Basic RNNs**: Trained via truncated BTTP.

**LSTM:** Offers great performance and faster training.

# Summary of today's lectures

RF: An ensemble of DTs, each trained on the random subspace method




A key hyperparameter: **"max_features"**

A measure for *interpretation*: **Feature importance**

# Question

So far: Learned about **DNNs, CNNs, RNNs** & **RFs.**

What if still unsatisfactory performances?

This may be due to:

1. $n \gg m \longleftarrow$ # of examples and/or

   data dimension

2. data distribution is pretty wide.

   i.e., data characteristics are quite distinct across examples.

# Techiques for addressing such scenarios

**Scenario 1:** $n \gg m$

⇨ dimensionality reduction

**Scenario 2:** data distribution is pretty wide.

⇨ clustering

# Outline of Day 6 lectures

Will study dimensionality reduction & clustering:

1. Explore the most popular dimensiona reduction technique: Principal Component Analysis (**PCA**)

2. Investigate another prominent technique:

   t-distributed Stochastic Neighbor Embedding (**t-SNE**)

3. Study clustering methods.