Today – Outline

- **Supervised** Learning basics:
  - Linear **regression**
  - Polynomial regression
- Lots of **Vocabulary**, notations
- Optimization basics: **Gradient Descent**
- **Supervised** Learning
  - **Classification** : Perceptron
  - multiple Loss functions, activations functions
- Optimisation strategies
- **Multi-class classification**

Multi-class classification

- If there are $K>2$ **classes.** Various strategies:

  One-versus-rest *(OVR)* strategy:

- Return to Binary Classif., a point is either class k or “not class k ”.

  → You now have K classifiers $W_{K,d} = \{\vec{w}_1, \ldots, \vec{w}_K\}$

- You have K times more parameters !

- Which one to choose ?
  The one that is the most on the correct side of the hyperplane:

$$\hat{y}_n = \text{argmax}_k(f_\Theta(\vec{x}_n)) = \text{argmax}_k(\vec{w}_k \vec{x}_n)$$

- What is a good Loss ?

Multi-class classification

Building a good Model+Loss for a **Multiclass** Perceptron:

- **Encode** classes into ***one-hot vectors***
  Ground truth of type: $t_n = \vec{e}_k = (0, \ldots, 0, 1, 0, \ldots, 0)$
  Network output : $\vec{y}^{(n)} = (y_1^{(n)}, \ldots, y_K^{(n)})$
- Use $\boldsymbol{softmax}(z)$: $z \in \mathbb{R}^K, \mathrm{softmax}(\vec{z})_j = \frac{\exp(z_j)}{\sum_k \exp(z_k)}$
- Model: assume $W_{K,d} = \{\vec{w}_1, \ldots, \vec{w}_K\}$

$$(y_n)_j = \mathrm{softmax}(W_{K,d}\vec{x}_n)_j = \frac{\exp(\vec{w}_j \cdot \vec{x}_n)}{\sum_k \exp(\vec{w}_k \cdot \vec{x}_n)}$$

Trick: insert $z_k = \vec{w}_k\vec{x}_n$ or $z_j = \vec{w}_j\vec{x}_n$

- Readout: $\hat{y}_n = \mathrm{argmax}_k((y_n)_k) = \mathrm{argmax}_k(\vec{w}_k\vec{x}_n)$
- Loss ? : see exercise "*Multi class classification*" in TD
  or "*TP2.2-MultiClass-Classification.ipynb*"

Multi-class classification

Two classic strategies :

- OVR: ***one-versus-rest*** (K)
  How to choose the winner ? Take the max. (argmax).
- OVO: ***one-versus-one*** $(K(K\text{-}1)/2)$
  How to choose the winner ? Take the one with most votes, typically.

References:

- **Linear classifiers in general**:
  → Bishop book, page 179-196, section 4.1
- Loss Function J2 (MSE for classif)
  → Bishop book, page **184-18**6, section 4.1.3
  (**Least squares for classification**)
- **Perceptron:**
  → Bishop book, page **192-196**, section 4.1.7
  (The **perceptron algorithm**)
- **Multi-Layer Perceptron**: see the Deep Learning course, OPT4 (Caio Corro)

Key concepts

- **Classification**
- **Readout** (vs activation function)
- **Model** vs Prediction (without readout, with it)
- Non trivial **losses**
- **Activations** : ReLu, softmax, sigmoids, logistic
- **Strategies**: Online, SGD, **mini-batch, full batch**
- Hyperplanes, Linearly Separable / non linearly separable data
- Encoding, **one-hot** vectors
- **Multi-class** Classification, **OVR** and OVO strategies