Primo parziale

Esercizio 1

1. **Multiple Choice:** Select the correct answer from the list of choices.

   (a) [5 points] True or False: Using the kernel trick, we can get non-linear decision boundaries using algorithms designed originally for linear models. ◯ True ◯ False

   (b) [5 points] True or False: A zero-mean Gaussian Prior (i.e. $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \sigma I)$) on model parameters in a MAP estimate corresponds to adding an L2 regularization term (e.g. $||\mathbf{w}||_2$) to the loss in an MLE estimate. ◯ True ◯ False

   (c) [5 points] True or False: In the Primal Form of the SVM, increasing hyperparameter C will decrease the complexity of the resulting classifier. ◯ True ◯ False

   (d) [5 points] True or False: The Maximum a Priori and Maximum likelihood solution for linear regression are always equivalent. ◯ True ◯ False

   (e) [5 points] If a hard-margin support vector machine tries to minimize $||\mathbf{w}||_2$ subject to $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 2$, what will be the size of the margin?

   ◯ $\frac{1}{||\mathbf{w}||}$ ◯ $\frac{2}{||\mathbf{w}||}$ ◯ $\frac{1}{2||\mathbf{w}||}$ ◯ $\frac{1}{4||\mathbf{w}||}$

   (f) [5 points] The posterior distribution of B given A is:

   ◯ $P(B \mid A) = \frac{P(A|B)P(A)}{P(B)}$

   ◯ $P(B \mid A) = \frac{P(A,B)P(B)}{P(A)}$

   ◯ $P(B \mid A) = \frac{P(A|B)P(B)}{P(A)}$

   ◯ $P(B \mid A) = \frac{P(A|B)P(B)}{P(A,B)}$

   (g) [5 points] Let $\mathbf{w}^*$ be the solution obtained using unregularized least-squares regression. What solution will you obtain is you scale all input features by a factor of c before solving?

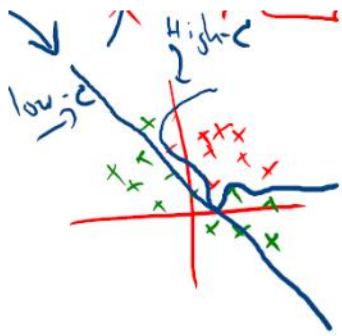
   ◯ $c\mathbf{w}^*$ ◯ $c^2\mathbf{w}^*$ ◯ $\frac{1}{c^2}\mathbf{w}^*$ ◯ $\frac{1}{c}\mathbf{w}^*$

Total Question 1: 35

a) Kernel trick , potrebbe essere vera [da specificare meglio anche se si intende non lineare nell'input space oppure nello spazio creato].
   Prof: Vero, "that's the point of doing the kernel trick". Quale è un altro modo per ottenere "non linear decision boundaries using linear algorithms?" Si può fare con qualunque non linear basis mapping [explicit embedding], come ad esempio utilizzando un mapping polinomiale. Il kernel trick fa questo implicitamente, utilizzando la Gram Matrix.
b) MAP [bayesian regression] vs MLE, dovrebbe essere vera. Differenze L1 e L2 nel caso della scelta dei pesi [sparse pesi che vuol dire].
   Prof: Vero. Vedi dimostrazione fatta in classe.
c) Falsa, quasi sicuro: aumentando C si porta il nostro modello ad aumentare la complessità perché si dà più peso ad ogni singola slack variable. Lezione 8 slide 26
   Prof: Falso, nella formulazione SVM la regolarizzazione è sugli errori di classificazione $slack$, quindi l'iperparametro C. Se aumentiamo C diciamo al modello di non preoccuparsi troppo del termine quadratico, ma di stare molto attento ai termini di slack.

$$\min_{\mathbf{w},b} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{n=1}^{N}\xi_n$$

$$\text{subject to} \quad y_n(\langle \mathbf{w}, \mathbf{x}_n\rangle + b) \geq 1 - \xi_n \text{ for all } n = 1, \ldots, N$$



d) MAP vs MLE, dovrebbe essere falsa perché non è vera <u>sempre</u> , in alcune condizioni è vero [vedi domanda b].
   Prof: Falso, MAP e MLE possono essere uguali se utilizziamo un regularizer L2, poniamo molta attenzione alla scelta del coefficiente di regolarizzazione along with the gaussian prior. Un facile modo per pensare a questo è il seguente: the MAP solution include il concetto di prior, il quale biasizza la soluzione per un specifico setting di model parameters. Ad esempio un gaussian prior a media nulla si ha bias della soluzione verso il vettore $\vec{0}$, che è la stessa cosa che fa la regolarizzazione L2.
e) $\frac{2}{||w||}$

prior a media nulla si ha bias della soluzione verso il vettore 0, che è la stessa cosa che fa la regolarizzazione L2.

e) $\frac{2}{\|w\|}$

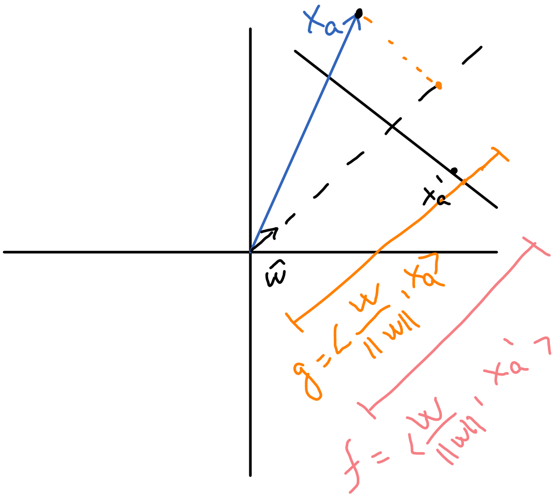
prova

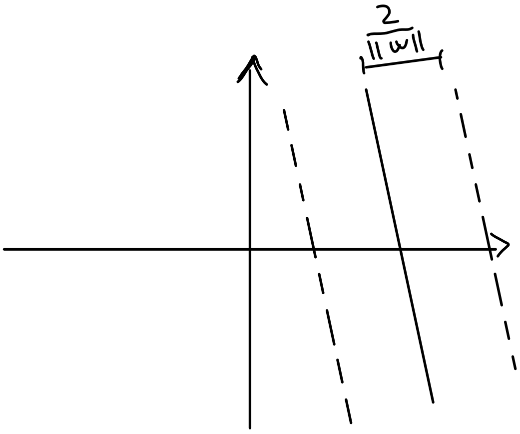
CIAO

portafortuna

$g = \langle \frac{w}{\|w\|}, x_a \rangle$

$f = \langle \frac{w}{\|w\|}, x_a' \rangle$





$y_n(\langle w, x_n \rangle + b) \geq 2$

f) E' semplicemente la regola di Bayes, quindi: $P(B|A) = \frac{P(A|B)P(B)}{P(A)}$

g) Prof: $\frac{1}{c}w^*$. Se scaliamo l'input per un valore c, allora avremo $c\vec{x}$. A questo punto per riottenere lo stesso risultato poiché $w^T\vec{x}$, si ha che i pesi dovranno essere riscalati per il termine $\frac{1}{c}$.

Idea nostra:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(\mathbf{w})\}^2$$

$$E = \frac{1}{2}\sum\{t_n - c\, w^T x\}^2$$

$$= \frac{1}{2}\sum\{t_n - c(w_0 + w_1 x)\}^2$$

$$\nabla E = \left[\frac{1}{2}\sum\{t_n - c(w_0 + w_1 x)\}^2 ; \frac{1}{2}\sum\{t_n - c(w_0 + w_1 x)\}^2\right]$$

$$\nabla_w E = \left[\frac{\frac{1}{2} \sum \{t_n - c(w_0 + w_1 x)\}^2}{\partial w_0} \quad ; \quad \frac{\frac{1}{2} \sum \{t_n - c(w_0 + w_1 x)\}^2}{\partial w_1} \right]$$

$$= \left[\sum \overbrace{(t_n - c(w_0 + w_1 x))}^{e_i} \cdot (-c) \qquad \sum \overbrace{(t_n - c(w_0 + w_1 x))}^{e_i} (-cx) \right]$$

$$= \sum e_i \left[-c \quad -cx \right] =$$

$$= -\sum c \cdot e_i \cdot \left[1 \quad x \right] = -c \sum e_i \cdot \left[1 \quad x \right]$$

Esercizio 2

2. Multiple Answer Select ALL correct choices: there may be more than one correct choice, but there is always at least one correct choice.

(a) [5 points] What are support vectors?

- ◯ The examples $\mathbf{x}_n$ from the training set required to compute the decision function $f(\mathbf{x})$ in an SVM.
- ◯ The class means.
- ◯ The training samples farthest from the decision boundary.
- ◯ The training samples $\mathbf{x}_n$ that are on the margin (i.e. $y_n f(\mathbf{x}_n) = 1$).

1

(b) [5 points] Which of the following are true about the relationship between the MAP and MLE estimators for linear regression?

- ◯ They are equal if $p(\mathbf{w}) = 1$
- ◯ They are equal if $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \sigma)$ for very small σ.
- ◯ They are never equal.
- ◯ They are equal in the limit of infinite training samples.

(c) [5 points] You train a linear classifier on 10,000 training points and discover that the training accuracy is only 67%. Which of the following, done in isolation, has a good chance of improving your training accuracy?

- ◯ Add novel features.
- ◯ Train on more data.
- ◯ Train on less data
- ◯ Regularize the model

(d) [5 points] In a soft-margin support vector machine, if we increase C, which of the following are likely to happen?

- ◯ The margin will grow wider.
- ◯ Most nonzero slack variables will decrease.
- ◯ $||\mathbf{w}||_2$ will grow larger.
- ◯ There will be more points inside the margin.

(e) [5 points] Which of the following are reasons why you might adjust your model in ways that increase the bias?

- ◯ You observe high training error and high validation error.
- ◯ You have few data points.
- ◯ You observe low training error and high validation error.
- ◯ Your data are not linearly separable.

(f) [5 points] Which of the following are true of polynomial regression (i.e. least squares regression with polynomial basis mapping)?

- ◯ If we increase the degree of polynomial, we increase variance.
- ◯ The regression function is nonlinear in the model parameters.
- ◯ The regression function is linear in the original input variables.
- ◯ If we increase the degree of polynomial, we decrease bias.

(g) [5 points] Which of the following classifiers can be used on non linearly separable datasets?

- ◯ The hard margin SVM.
- ◯ Logistic regression.
- ◯ The linear generative Bayes classifiers.
- ◯ Fisher's Linear Discriminant.

Esercizio 3

3. [15 points] Assume the class conditional distributions for a two-class classification problem are $p(\mathbf{x} \mid \mathcal{C}_1) = \mathcal{N}(\boldsymbol{\mu}_1, \beta^{-1}I)$ and $p(\mathbf{x} \mid \mathcal{C}_2) = \mathcal{N}(\boldsymbol{\mu}_2, \beta^{-1}I)$. Show that the optimal decision boundary is *linear*, i.e. that it can be written as $H = \{\mathbf{x} \mid \mathbf{w}^T\mathbf{x} + b = 0\}$ for some $\mathbf{w}$ and b.

   Hint: Remember that points $\mathbf{x}$ on the optimal decision boundary will satisfy $p(\mathcal{C}_1 \mid \mathbf{x}) = p(\mathcal{C}_2 \mid \mathbf{x})$, and that the formula for the multivariate Gaussian density is:

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\}$$

Probabile soluzione

Esercizio 4

4. [15 points] Assume we have a training set of only two points (one from each class):

$$\mathcal{D} = \{([0,0], -1), ([2,0], +1)\}$$

Solve for the optimal hard margin primal SVM parameters $\mathbf{w}$ and b for this dataset.

HM SVM: $\min_{w,b} \frac{1}{2} \|w\|^2$

con vincoli $y_n (\langle w, x_n \rangle + b) \geq 1$

Poiché gli unici due punti che abbiamo sono support vectors, allora si ha che saranno sui [rispettivi] margini, quindi si può scrivere:

$$y_n (\langle \vec{w}, \vec{x}_n \rangle + b) = 1$$

Mettiamo i punti che abbiamo:

- $-1 \left([w_0 \quad w_1] \begin{bmatrix} 0 \\ 0 \end{bmatrix} + b \right) = 1 \Rightarrow -1 (0+b) = 1 \Rightarrow b = -1$
- $1 \left([w_0 \quad w_1] \begin{bmatrix} 2 \\ 0 \end{bmatrix} + b \right) = 1 \Rightarrow 2 w_0 + b = 1 \Rightarrow w_0 = 1$

$$w^* = [1 \quad w_1] \qquad b = -1$$

Dovendo minimizzare $\|w\|$ e non avendo vincoli sulla scelta di w_1 si procede a porre $w_1 = 0$. Quindi:

$$w^* = [1 \quad 0] \qquad b = -1$$

Esercizio 5

5. [10 points (bonus)] Write the primal SVM objective as an empirical risk minimization problem over a linearly separable dataset $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$. Show that any $\mathbf{w}$ and b minimizing this empirical risk is also a solution to the standard hard-margin SVM objective with constraints.

Argomento collegato

Il problema di empirical risk minization:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C \sum_{n=1}^{N} \max\{0, 1 - y_n(\langle w, x_n \rangle + b)\}$$

Il problema di hard-margin SVM:

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

con vincoli $\quad y_n(\langle w, x_n \rangle + b) \geq 1$

Poiché il dataset è linearmente separabile si ha che hard margin ammette una soluzione, in cui tutti i sample verrano classificati correttamente. Questo porta ad avere che $\forall (x_n, y_n)$ si ha $y_n(< w, x_n > +b) \geq 1$. Mettendo questa osservazione nel termine di hinge loss avremo che $\max\{0, 1 - y_n(< w, x_n > +b)\}$ dove $1 - y_n(< w, x_n > +b) \leq 0$ e quindi tale termine di loss sarà zero. Quindi i w e b che saranno soluzione hard-margin saranno soluzione anche per empirical risk.

Secondo parziale

Esercizio 1

1. **Multiple Choice**: Select the correct answer from the list of choices.

   (a) [5 points] True or False: A K-nearest neighbor classifier is only able to learn linear discriminant functions. ○ True ☒ False

   (b) [5 points] True or False: Projecting a dataset onto its first principal component maximizes the variance of the projected data. ☒ True ○ False

   (c) [5 points] True or False: The K-means algorithm is guaranteed to find the best cluster centers for any dataset. ○ True ☒ False

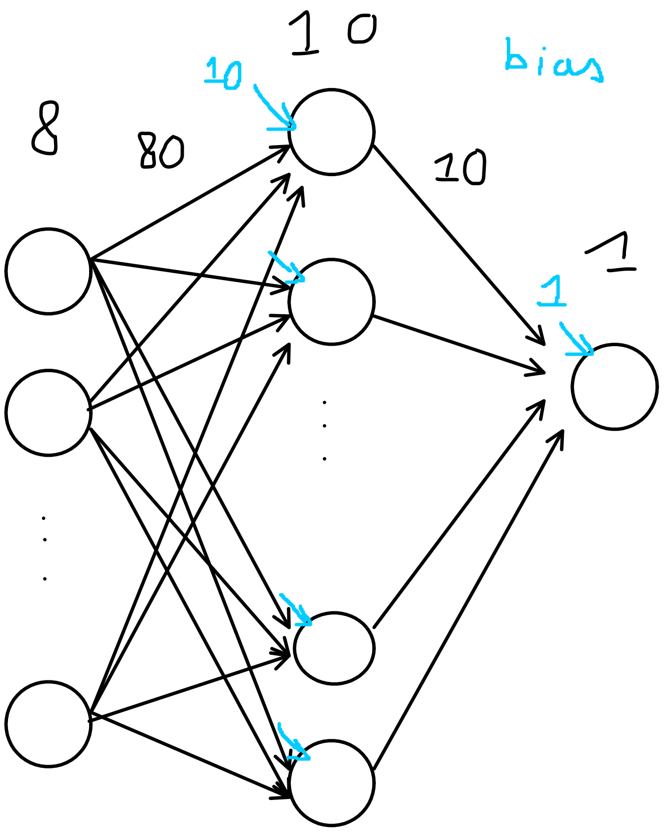
   (d) [5 points] True or False: A Parzen kernel density estimator uses only the nearest sample in the dataset to estimate the probability of an input sample x. ○ True ☒ False

   (e) [5 points] How many parameters will a Multilayer Perceptron (MLP) for binary classification with a single hidden layer of width 10 and an input dimensionality of 8 have?
   ○ 80 ○ 99 ○ 88 ☒ None of the above 10

   (f) [5 points] Which of the following loss functions is called the negative log likelihood?
   - ○ $\mathcal{L}(\mathrm{y}, \hat{\mathrm{y}}) = -\sum_{c=1}^{C} (\ln y_c - \ln \hat{y}_c)^2$
   - ○ $\mathcal{L}(\mathrm{y}, \hat{\mathrm{y}}) = -\sum_{c=1}^{C} (y_c - \ln \hat{y}_c)^2$
   - ☒ $\mathcal{L}(\mathrm{y}, \hat{\mathrm{y}}) = -\sum_{c=1}^{C} y_c \ln \hat{y}_c$
   - ○ $\mathcal{L}(\mathrm{y}, \hat{\mathrm{y}}) = -\sum_{c=1}^{C} \ln \hat{y}_c$

   (g) [5 points] How many iterations of gradient descent must we perform for an epoch of minibatch Stochastic Gradient Descent with a dataset of 1024 samples and a batch size of 16?
   ○ 1024 ○ 1 ○ 32 ☒ 64

Total Question 1: 35



tot = 101

1. **Multiple Choice:** Select the correct answer from the list of choices.

   (a) [5 points] True or False: A K-nearest neighbor classifier is only able to learn linear discriminant functions. ○ True √ **False**

   (b) [5 points] True or False: Projecting a dataset onto its first principal component maximizes the variance of the projected data. √ **True** ○ False

   (c) [5 points] True or False: The K-means algorithm is guaranteed to find the best cluster centers for any dataset. ○ True √ **False**

   (d) [5 points] True or False: A Parzen kernel density estimator uses only the nearest sample in the dataset to estimate the probability of an input sample $\mathbf{x}$. ○ True √ **False**

   (e) [5 points] How many parameters will a Multilayer Perceptron (MLP) for binary classification with a single hidden layer of width 10 and an input dimensionality of 8 have?

   ○ 80 √ **99** ○ 88 ○ None of the above

   (f) [5 points] Which of the following loss functions is called the negative log likelihood?

   ○ $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c=1}^{C}(\ln y_c - \ln \hat{y}_c)^2$

   ○ $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c=1}^{C}(y_c - \ln \hat{y}_c)^2$

   √ $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c=1}^{C} y_c \ln \hat{y}_c$

   ○ $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{c=1}^{C} \ln \hat{y}_c$

   (g) [5 points] How many iterations of gradient descent must we perform for an epoch of minibatch Stochastic Gradient Descent with a dataset of 1024 samples and a batch size of 16?

   ○ 1024 ○ 1 ○ 32 √ **64**

Total Question 1: 35

Esercizio 2

(a) [5 points] What are the advantages of projecting data onto $K < D$ principal components?

- ◯ We eliminate noise in the original representation.
- ◯ Classes are guaranteed to be linearly separable.
- ◯ It is a nonlinear embedding that makes learning easy with simpler models.
- ◯ Models trained on the reduced data are simpler.

(b) [5 points] Which of the following are advantages of Ensemble Models (e.g. Committees)?

- ◯ They reduce the variance of the resulting model.
- ◯ They are much more efficient than the base model.
- ◯ They can reduce the expected error of the final model.
- ◯ The resulting model is nonlinear even if the base model is linear.

(c) [5 points] Which of the following are causes of the vanishing gradients when training neural networks?

- ◯ Saturated inputs to activation functions with near-zero derivatives when saturated.
- ◯ Badly scaled input values.
- ◯ Very deep models.
- ◯ Bad random initialization of the network parameters.

(d) [5 points] Which of the following are requirements for applying backpropagation to compute gradients in a deep network?

- ◯ The network must not be too deep.
- ◯ The network must be a directed acyclic graph.
- ◯ All activation functions must be differentiable.
- ◯ All activation functions must be continuous.

(e) [5 points] Which of the following are true of the Nadaraya-Watson estimator?

- ◯ It only requires some of the training data at test time.
- ◯ It is a nonparametric method.
- ◯ It estimates a nonlinear function of the input.
- ◯ It estimates a linear function of the input.

(f) [5 points] What does the learning rate control in Stochastic Gradient Descent?

- ◯ The size of gradient steps made in each iteration.
- ◯ The degree of nonlinearity in the model.
- ◯ ~~The regression function is linear in the original input variables.~~
- ◯ The speed at which the model learns.

(g) [5 points] Which of the following models are nonparametric?

- ◯ The Multilayer Perceptron (MLP).
- ◯ Logistic regression.
- ◯ The K-Nearest Neighbor Classifier
- ◯ Decision Trees.

(a) [5 points] What are the advantages of projecting data onto $K < D$ principal components?

- $\surd$ **We eliminate noise in the original representation.**
- ◯ Classes are guaranteed to be linearly separable.
- ◯ It is a nonlinear embedding that makes learning easy with simpler models.
- $\surd$ **Models trained on the reduced data are simpler.**

(b) [5 points] Which of the following are advantages of Ensemble Models (e.g. Committees)?

- $\surd$ **They reduce the variance of the resulting model.**
- ◯ They are much more efficient than the base model.
- $\surd$ **They can reduce the expected error of the final model.**
- ◯ The resulting model is nonlinear even if the base model is linear.

(c) [5 points] Which of the following are causes of the vanishing gradients when training neural networks?

- $\surd$ **Saturated inputs to activation functions with near-zero derivatives when saturated.**
- ◯ Badly scaled input values.
- $\surd$ **Very deep models.**
- ◯ Bad random initialization of the network parameters.

(d) [5 points] Which of the following are requirements for applying backpropagation to compute gradients in a deep network?

- ◯ The network must not be too deep.
- $\surd$ **The network must be a directed acyclic graph.**
- $\surd$ **All activation functions must be differentiable.**
- ◯ All activation functions must be continuous.

(e) [5 points] Which of the following are true of the Nadaraya-Watson estimator?

- ◯ It only requires some of the training data at test time.
- $\surd$ **It is a nonparametric method.**
- $\surd$ **It estimates a nonlinear function of the input.**
- ◯ It estimates a linear function of the input.

(f) [5 points] What does the learning rate control in Stochastic Gradient Descent?

- $\surd$ **The size of gradient steps made in each iteration.**
- ◯ The degree of nonlinearity in the model.
- ◯ The regression function is linear in the original input variables.
- $\surd$ **The speed at which the model learns.**

(g) [5 points] Which of the following models are nonparametric?

- ◯ The Multilayer Perceptron (MLP).
- ◯ Logistic regression.
- $\surd$ **The K-Nearest Neighbor Classifier**
- ◯ Decision Trees.

Total Question 2: 35

Esercizio 3

3. [15 points] Show that the first principal component of dataset $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^{N}$ is an eigenvector of the data covariance matrix.

$$S = \frac{1}{N} \sum_{n=1}^{N} (\vec{x} - \bar{\vec{x}})(\vec{x} - \bar{\vec{x}})^T$$

$$\Downarrow$$

scalare $\qquad$ scalare

$$\vec{u}_1^T S \vec{u}_1 = \frac{1}{N} \sum_{n=1}^{N} (\vec{u}_1 \cdot \vec{x} - \vec{u}_1 \cdot \bar{\vec{x}})^2$$

$$\Downarrow$$

devo massimizzare la varianza nello spazio proiettato però devo vincolare $\vec{u}_1$ (lagrangiano)

$$\Downarrow$$

nuova f. obj

$$\vec{u}_1^T S \vec{u}_1 + \lambda_1 (1 - \vec{u}_1^T \vec{u}_1)$$

$\Downarrow$ massimizzo rispetto alla direzione

$$\nabla_{\vec{u}_1} \vec{u}_1^T S \vec{u}_1 + \lambda_1 (1 - \vec{u}_1^T \vec{u}_1) = 0$$

$$\Downarrow$$

$$S \vec{u}_1 - \lambda_1 \vec{u}_1 = 0$$

$$\Downarrow$$

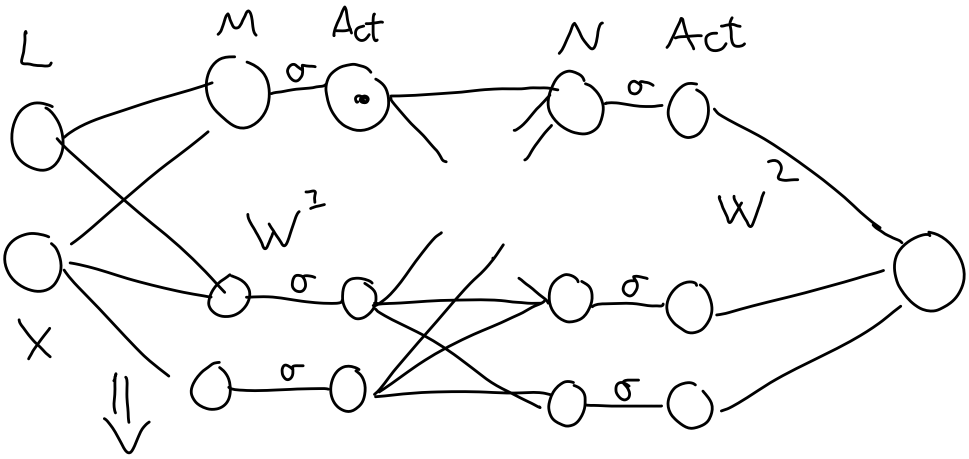
$$S \vec{u}_1 = \lambda_1 \vec{u}_1$$

$\Downarrow$ moltiplico per $\vec{u}_1^T$

$$\vec{u}_1^T S \vec{u}_1 = \lambda_1$$

Esercizio 4

4. [15 points] Show that a Multilayer Perceptron with two hidden layers with activation function $\sigma(x) = x$ is only capable of learning linear functions.



$$Y = \sigma\left(W_2\,\sigma(W_1 x + b_1) + b_2\right) = f(x)$$

$$f(cx) = c\,f(x)$$

$$Y = f\left(\sigma\left(W_2\,\sigma(W_1 x)\right)\right)$$

$$\sigma(x) = x$$

$$Y = f\left(\sigma\left(W_2(W_1 x)\right)\right) \Rightarrow y = f(W_2 W_1 x)$$

$$x \to ax \Rightarrow y = f\left(W_2 W_1 (ax)\right) = a\,(W_2 W_1 x)$$

Solution: An MLP with two hidden layers computes the function:

$$\begin{aligned} f(\mathbf{x}) &= W_{\text{out}}\sigma(W_2\sigma(W_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_{\text{out}} \\ &= W_{\text{out}}(W_2(W_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_{\text{out}} \text{ (since } \sigma \text{ is the identity function)} \\ &= (W_{\text{out}}W_2W_1)\mathbf{x} + [W_{\text{out}}W_2\mathbf{b}_1 + W_{\text{out}}\mathbf{b}_2 + \mathbf{b}_{\text{out}}], \end{aligned}$$

which is a linear (well, affine) function $f(\mathbf{x}) = W\mathbf{x} + \mathbf{b}$ for:

$$\begin{aligned} W &= W_{\text{out}}W_2W_1 \\ \mathbf{b} &= W_{\text{out}}W_2\mathbf{b}_1 + W_{\text{out}}\mathbf{b}_2 + \mathbf{b}_{\text{out}}. \end{aligned}$$

□

Esercizio 5

5. [10 points (bonus)] Design a Deep Convolutional Neural Network (with at least three convolutional layers one or more pooling layers) to classify MNIST images (input size 28×28). Draw the network (or write pseudocode for its definition) and indicate how many parameters each layer has and the sizes of the intermediate feature maps.