

Towards Simple and Effective Connectionist Nonparametric Estimation of Probability Density Functions

Edmondo Trentin

Dipartimento di Ingegneria dell'Informazione
Università di Siena, V. Roma, 56 - Siena (Italy)
trentin@dii.unisi.it

December 14, 2006



[Title Page](#)



Page 1 of 19

[Go Back](#)

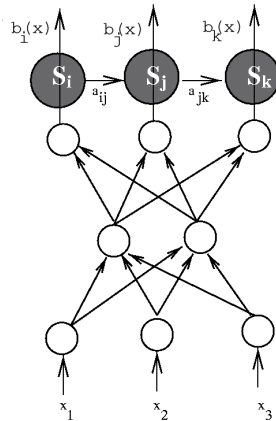
[Full Screen](#)

[Close](#)

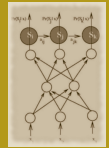
[Quit](#)

Prologue: hybrid ANN/HMM model

- HMM topology with standard *initial* and *transition* probabilities a_{ij} for each pair of states i, j
- The i -th ANN output $o_i(t)$ at time t represents the **emission probability** $b_{i,t}(\mathbf{y}_t)$



- Recognition is accomplished via *Viterbi*, while a **maximum-likelihood (ML) global training technique** is used



Title Page



Page 2 of 19

Go Back

Full Screen

Close

Quit

Prologue 2: Learning Algorithm

Criterion function: *likelihood* $L = \sum_{i \in \mathcal{F}} \alpha_{i,T}$.

Learning rule:

$$\Delta w_{jk} = \eta \frac{\partial L}{\partial w_{jk}} \quad (1)$$

Defining

$$\delta_j(i, t) = \begin{cases} f'_j(x_j(t)) & \text{if } l = L, i = j \\ 0 & \text{if } l = L, i \neq j \\ f'_j(x_j(t)) \sum_{n \in \mathcal{L}_{l+1}} w_{ni} \delta_n(j, t) & \text{otherwise} \end{cases} \quad (2)$$

for each $i \in \mathcal{L}_l$, and after *Bridle 1990*, *Bengio 1992* we have

$$\Delta w_{jk} = \eta \sum_{i=1}^Q \sum_{t=1}^T \beta_{i,t} \frac{\alpha_{i,t}}{b_{i,t}} \delta_j(i, t) o_k(t). \quad (3)$$



Title Page



Page 3 of 19

Go Back

Full Screen

Close

Quit

Prologue 3: the “Divergence Problem”

- **Problem:** the ML criterion leads to degenerate solutions, since the ANN is not *constrained to be a pdf*
- **Empirical/theoretical** solutions were proposed in the ANN/HMM scenario (e.g., architectural, Soft-weight sharing, Maximum-A-Posteriori, etc.)
- We felt the need for **ANNs that estimate pdfs**

(END OF PROLOGUE, BEGINNING OF TALK)



Title Page



Page 4 of 19

Go Back

Full Screen

Close

Quit

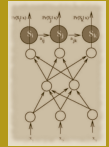
General setup: unsupervised learning

1. Topological framework

- (a) Clustering algorithms
- (b) Competitive Neural Nets
- (c) Self-organizing Maps

2. Probabilistic framework: estimation of probability density functions (pdf)

- (a) Parametric techniques (maximum-likelihood, min-max): arbitrary assumption on the form of the underlying distribution
- (b) **Nonparametric techniques** (Parzen Window, k_n -Nearest Neighbor): direct estimation of the pdf from a data sample



Title Page



Page 5 of 19

Go Back

Full Screen

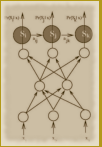
Close

Quit

Pdf estimation setup

- Let us consider a pdf $p(\mathbf{x})$
- Let \mathbf{x}' be drawn from $p(\mathbf{x})$
- Let R be an arbitrary region of the d -dimensional feature space
- Unsupervised sample: $\mathcal{T} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of n patterns, i.i.d. according to $p(\mathbf{x})$

Goal: assuming $p(\cdot)$ is unknown, estimate $p(\mathbf{x}')$ relying on \mathcal{T} .



Title Page



Page 6 of 19

Go Back

Full Screen

Close

Quit

Parzen Window (PW) method

(a) $P(\mathbf{x}' \in R) = \int_R p(\mathbf{x}) d\mathbf{x}$

(b) If k_n patterns in \mathcal{T} fall within R , then

$$P(\mathbf{x}' \in R) \simeq k_n/n$$

(c) Since $\int_R p(\mathbf{x}) d\mathbf{x} \simeq p(\mathbf{x}')V$ (where V is the volume of region R) we have the PW estimate:

$$\begin{aligned} p(\mathbf{x}') &\simeq \frac{k_n/n}{V} \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{V} \varphi\left(\frac{\mathbf{x}' - \mathbf{x}_i}{h}\right) \end{aligned}$$

where $\varphi(\cdot)$ is a zero-centered **window function** having edge h , i.e. $V = h^d$.



Title Page



Page 7 of 19

Go Back

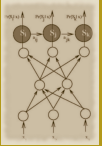
Full Screen

Close

Quit

PW limitations

- (i) the estimate is not in a compact functional form; it is a **sum of as many local windows as the size of the sample**;
- (ii) the local nature of the window functions yields a fragmented model (“**memory based**”, prone to overfitting);
- (iii) **high complexity**: the whole training sample has to be kept always in memory;
- (iv) **the form of the window function chosen has a deep influence** on the form of the estimated model;
- (v) the PW model heavily **depends on the choice of an initial width** of the window functions.



Title Page



Page 8 of 19

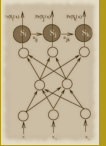
Go Back

Full Screen

Close

Quit

Artificial neural networks (ANN) for pdf estimation



- ANNs are an alternative family of **nonparametric models**
- “**Universal approximation**” property of certain ANN families (MLP, RBF)
- ANNs are used for estimating *probabilities* (easy task)
- ANNs **have not been exploited so far for estimating pdfs** (unsupervised and far less obvious task)

Title Page



Page 9 of 19

Go Back

Full Screen

Close

Quit

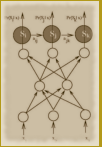
Parzen Neural Network (PNN)

Algorithm: train a feedforward ANN in order to learn $p(\mathbf{x})$ from the unsupervised dataset \mathcal{T} .

Input: $\mathcal{T} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, h_1 .

Output: $\hat{p}(\cdot)$ /* estimate of $p(\cdot)$ */

1. Let $h = h_1 / \sqrt{n}$
2. Let $V = h^d$
3. For $i=1$ to n do
 - 3.1 Let $\mathcal{T}_i = \mathcal{T} \setminus \{\mathbf{x}_i\}$
 - 3.2 Let $y_i = \frac{1}{n-1} \sum_{\mathbf{x} \in \mathcal{T}_i} \frac{1}{V} \varphi\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)$
4. Let $\mathcal{S} = \{(\mathbf{x}_i, y_i) \mid i = 1, \dots, n\}$
5. Train ANN via backprop over \mathcal{S}
6. Let $\hat{p}(\cdot)$ be equal to the ANN
7. Return $\hat{p}(\cdot)$



Title Page



Page 10 of 19

Go Back

Full Screen

Close

Quit

Note that:

1. The **PNN output should be non-negative**, but it may take values in the $[0, +\infty)$ range:
 - (a) $y = \frac{\lambda}{1+e^{-x}}$ with trainable λ (*Trentin, 2001*)
 - (b) linear output activation functions, forcing negative outputs to zero
2. As in the k_n -nearest neighbor technique, the **PNN is not necessarily a pdf** (in general, $\int \hat{p}(\mathbf{x})d\mathbf{x} \neq 1$)
3. The PW generation of target outputs (steps 3-3.2) is “**unbiased**”.
4. The PNN is trained only over the locations (in the feature space) of the patterns belonging to the original sample.



Title Page



Page 11 of 19

Go Back

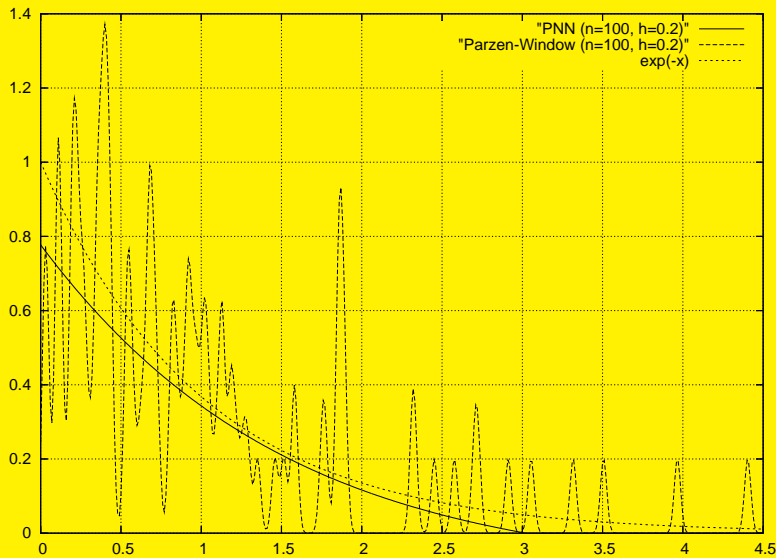
Full Screen

Close

Quit

Illustrative estimation task

Exponential distribution: $p(x) = e^{-x}$, $x \geq 0$



$n = 100$ points drawn from $p(x)$, $h_1 = 0.2$.



Title Page



Page 12 of 19

Go Back

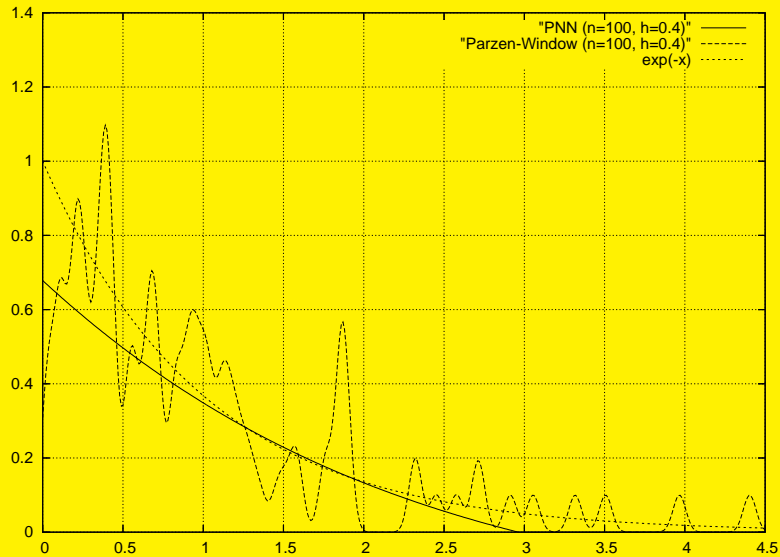
Full Screen

Close

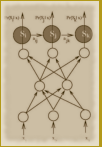
Quit

Illustrative estimation task

Exponential distribution: $p(x) = e^{-x}$, $x \geq 0$



$n = 100$ points drawn from $p(x)$, $h_1 = 0.4$.



Title Page



Page 13 of 19

Go Back

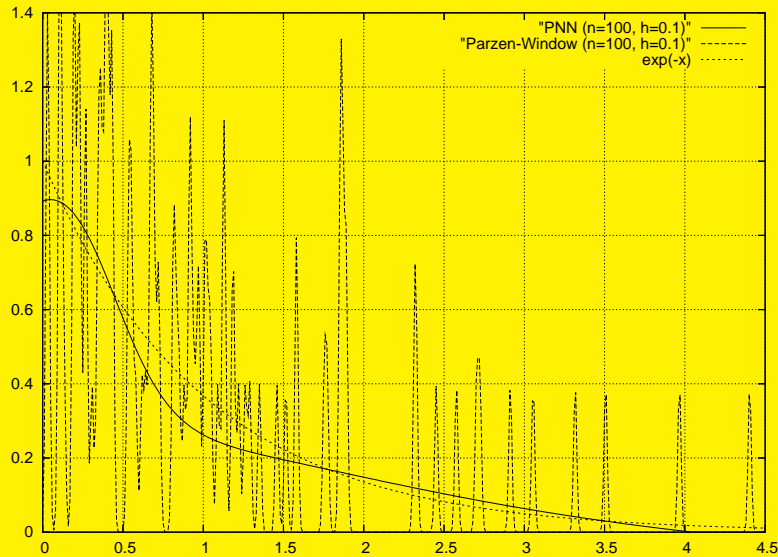
Full Screen

Close

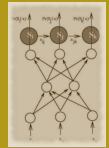
Quit

Illustrative estimation task

Exponential distribution: $p(x) = e^{-x}$, $x \geq 0$



$n = 100$ points drawn from $p(x)$, $h_1 = 0.1$.



Title Page



Page 14 of 19

Go Back

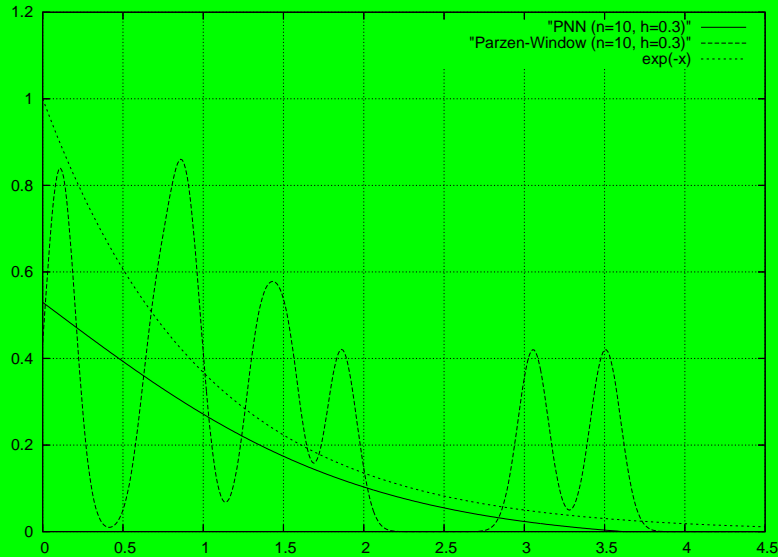
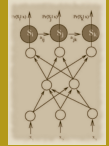
Full Screen

Close

Quit

Varying the sample size

$$n = 10$$



$$(h_1 = 0.3)$$

Title Page



Page 15 of 19

Go Back

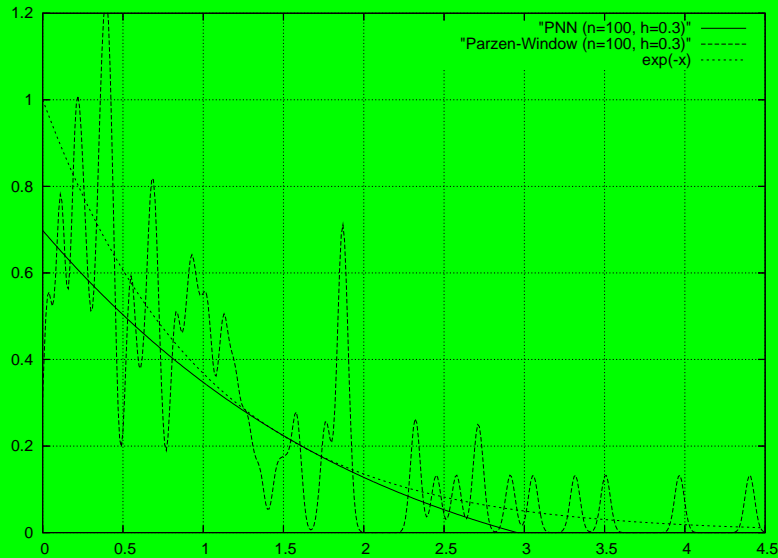
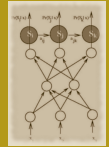
Full Screen

Close

Quit

Varying the sample size

$$n = 100$$



$$(h_1 = 0.3)$$

Title Page



Page 16 of 19

Go Back

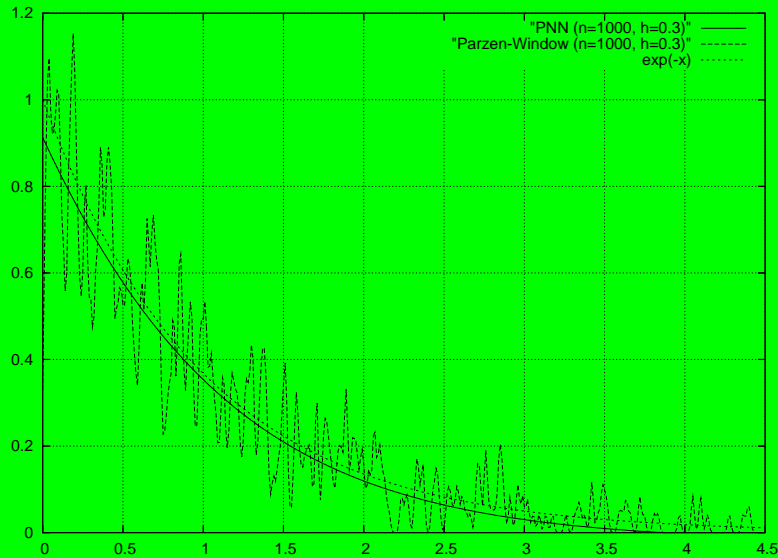
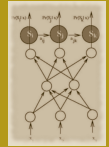
Full Screen

Close

Quit

Varying the sample size

$$n = 1000$$



$$(h_1 = 0.3)$$

Title Page



Page 17 of 19

Go Back

Full Screen

Close

Quit

Bioinformatics (preliminary results)

Task: classification of the **bond state of cysteines** within amino acid sequences of the PDB (Protein DataBank), relying on the primary structure of proteins.

- **Data:** Jan 2001 version of PDB, **967 sequences**, 20-fold leave-one-out according to *Frasconi et al.*
- **Feature space:** 15-aminoacid windows (centered on cysteine) over the profile of multiple alignment (i.e., **300-dim patterns**)
- **Results:** PW: 76.11% MLP: 80.25% PNN: 81.16%

(*Note: state-of-the-art is approx. 88% by Frasconi et al.*)



Title Page



Page 18 of 19

Go Back

Full Screen

Close

Quit

(Preliminary) Conclusions

- Simple and effective gradient-based *unsupervised training of feed-forward ANNs*
- “Universal” non-parametric estimation of *pdfs*
- PNN overcomes major PW limitations:
 - (i) the estimate *is* in a compact functional form
 - (ii) PNN is not “memory based”: a general “law” is inferred from the data
 - (iii) reduced complexity: the PNN may be small, the training sample has *not* to be kept always in memory
 - (iv) generalization capabilities of the PNN are basically independent from the form of the window function
 - (v) the PNN is less sensitive to the initial choice of *h*



Title Page



Page 19 of 19

Go Back

Full Screen

Close

Quit