Machine Intelligence II

Online-PCA (Hebbian Learning)

Tutorial

Neural Information Processing Group (Prof. Dr. Klaus Obermayer)

19.05.2016

Projection Methods

- Principal Component Analysis (PCA)
- Online-PCA $\Rightarrow$ Hebbian Learning
- Nonlinear Structure $\Rightarrow$ Kernel PCA
- Source Separation
  - Model based $\Rightarrow$ Independent Component Analysis (ICA)
  - Cost Function Based $\Rightarrow$ Projection Pursuit

Principal Components (PCs): "interesting" directions

Eigenvalue problem yields direction in feature space with max. variance

$$\underline{\mathbf{C}}\mathbf{e}_i = \lambda_i \underline{\mathbf{e}}_i, \qquad i = 1, \ldots, N$$

Principal Components (PCs): "interesting" directions

Eigenvalue problem yields direction in feature space with max. variance

$$\underline{\mathbf{C}}\underline{\mathbf{e}}_i = \lambda_i \underline{\mathbf{e}}_i, \qquad i = 1, \ldots, N$$

$\Rightarrow$ **Principal Components**: normalized eigenvector $\underline{\mathbf{e}}_i$ of $\underline{\mathbf{C}}$

$\Rightarrow$ Variance along a PC is given by the corresponding Eigenvalue

$$\sigma_i^2 = \underline{\mathbf{e}}_i^T \underline{\mathbf{C}}\underline{\mathbf{e}}_i = \lambda \underline{\mathbf{e}}_i^2 = \lambda_i$$

Principal Components (PCs): "interesting" directions

Eigenvalue problem yields direction in feature space with max. variance

$$\underline{\mathbf{C}}\underline{\mathbf{e}}_i = \lambda_i \underline{\mathbf{e}}_i, \qquad i = 1, \ldots, N$$

⇒ **Principal Components**: normalized eigenvector $\underline{\mathbf{e}}_i$ of $\underline{\mathbf{C}}$

⇒ Variance along a PC is given by the corresponding Eigenvalue

$$\sigma_i^2 = \underline{\mathbf{e}}_i^T \underline{\mathbf{C}}\underline{\mathbf{e}}_i = \lambda \underline{\mathbf{e}}_i^2 = \lambda_i$$

⇒ ordering of principal components w.r.t. variance:

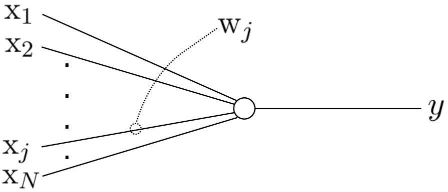
$$\begin{array}{ccccccccccc} \lambda_1 & > & \lambda_2 & > & \lambda_3 & > & \ldots\ldots & > & \lambda_{N-1} & > & \lambda_N \\ \downarrow & & \downarrow & & \downarrow & & & & \downarrow & & \downarrow \\ \underline{\mathbf{e}}_1 & & \underline{\mathbf{e}}_2 & & \underline{\mathbf{e}}_3 & & & & \underline{\mathbf{e}}_{N-1} & & \underline{\mathbf{e}}_N \end{array}$$

$\underline{\mathbf{e}}_i$: direction of largest variance in the subspace $\mathrm{span}\{\underline{\mathbf{e}}_i, \underline{\mathbf{e}}_{i+1}, \ldots, \underline{\mathbf{e}}_N\}$

Online-PCA

Biologically plausible implementation
online learning

Linear connectionist neurons



$$y = \underline{\mathbf{w}}^T \underline{\mathbf{x}}$$

observations: $\underline{\mathbf{x}}^{(\alpha)}, \alpha = 1, \ldots, p, \underline{\mathbf{x}}^{(\alpha)} \in \mathbb{R}^N$

Hebbian (correlation-based) learning for linear neurons

The Organization of Behavior (Donald Hebb,1949): fire together-wire together

Hebbian (correlation-based) learning for linear neurons

The Organization of Behavior (Donald Hebb,1949): fire together-wire together

initialization of weights $\underline{\mathbf{w}}$ (e.g. sample small random numbers)
choose (small) learning rate ε
begin loop

Choose an observation $\underline{\mathbf{x}}^{(\alpha)}$
Change weights according to:

$$\Delta \mathrm{w}_j = \varepsilon y_{\left(\underline{\mathbf{x}}^{(\alpha)};\underline{\mathbf{w}}\right)} x_j^{(\alpha)} = \varepsilon \left(\underline{\mathbf{w}}^T \underline{\mathbf{x}}^{(\alpha)}\right) x_j^{(\alpha)}, \qquad j = 1, \ldots, N$$

end

Hebbian (correlation-based) learning for linear neurons

The Organization of Behavior (Donald Hebb,1949): fire together-wire together

initialization of weights $\underline{\mathbf{w}}$ (e.g. sample small random numbers)
choose (small) learning rate ε
begin loop

Choose an observation $\underline{\mathbf{x}}^{(\alpha)}$
Change weights according to:

$$\Delta \mathrm{w}_j = \varepsilon y_{\left(\underline{\mathbf{x}}^{(\alpha)};\underline{\mathbf{w}}\right)} x_j^{(\alpha)} = \varepsilon \left(\underline{\mathbf{w}}^T \underline{\mathbf{x}}^{(\alpha)}\right) x_j^{(\alpha)}, \qquad j = 1, \ldots, N$$

end

$\Rightarrow$ weights increase (decrease) if input and output are correlated (anticorrelated)

Hebbian (correlation-based) learning for linear neurons

The Organization of Behavior (Donald Hebb,1949): fire together-wire together

initialization of weights $\underline{\mathbf{w}}$ (e.g. sample small random numbers)
choose (small) learning rate ε
begin loop

Choose an observation $\underline{\mathbf{x}}^{(\alpha)}$
Change weights according to:

$$\Delta \mathrm{w}_j = \varepsilon y_{\left(\underline{\mathbf{x}}^{(\alpha)};\underline{\mathbf{w}}\right)} x_j^{(\alpha)} = \varepsilon \left(\underline{\mathbf{w}}^T \underline{\mathbf{x}}^{(\alpha)}\right) x_j^{(\alpha)}, \qquad j = 1, \ldots, N$$

end

$\Rightarrow$ weights increase (decrease) if input and output are correlated (anticorrelated)

Proposition

Hebb's rule extracts the PC with the largest eigenvalue: $\underline{\mathbf{w}} \propto \underline{\mathbf{e}}_1$

Proofs: deterministic version $\rightarrow$ lecture notes; stochastic version $\rightarrow$ Haykin book

Implications

Implications

- model for learning of receptive fields and the resulting *neural code*

Implications

- model for learning of receptive fields and the resulting *neural code*
- adaptive tracking of the direction of largest variance: "on-line" PCA

Implications

- model for learning of receptive fields and the resulting *neural code*
- adaptive tracking of the direction of largest variance: "on-line" PCA

Problem: $\|w\| \to \infty \Rightarrow$ requires some form of ("biological"?) normalization

Implications

- model for learning of receptive fields and the resulting *neural code*
- adaptive tracking of the direction of largest variance: "on-line" PCA

Problem: $\|w\| \rightarrow \infty \Rightarrow$ requires some form of ("biological"?) normalization

Solution: Normalization via Oja's rule

Implications

- model for learning of receptive fields and the resulting *neural code*
- adaptive tracking of the direction of largest variance: "on-line" PCA

Problem: $\|w\| \to \infty \Rightarrow$ requires some form of ("biological"?) normalization

Solution: Normalization via Oja's rule

$$\Delta \mathrm{w}_j = \varepsilon y_{\left(\underline{\mathbf{x}}^{(\alpha)};\underline{\mathbf{w}}\right)} \left\{ \underbrace{\mathrm{x}_j^{(\alpha)}}_{\substack{\text{Hebbian}\\ \text{learning}}} - \underbrace{y_{\left(\underline{\mathbf{x}}^{(\alpha)};\underline{\mathbf{w}}\right)} \mathrm{w}_j}_{\text{decay term}} \right\}$$

Implications

- model for learning of receptive fields and the resulting *neural code*
- adaptive tracking of the direction of largest variance: "on-line" PCA

Problem: $\|w\| \to \infty \Rightarrow$ requires some form of ("biological"?) normalization

Solution: Normalization via Oja's rule

$$\Delta \mathrm{w}_j = \varepsilon y_{\left(\underline{\mathbf{x}}^{(\alpha)};\underline{\mathbf{w}}\right)} \left\{ \underbrace{\mathrm{x}_j^{(\alpha)}}_{\substack{\text{Hebbian} \\ \text{learning}}} - \underbrace{y_{\left(\underline{\mathbf{x}}^{(\alpha)};\underline{\mathbf{w}}\right)} \mathrm{w}_j}_{\text{decay term}} \right\}$$

Proposition

Oja's rule converges to the unit vector which points into the direction of the largest variance: $\underline{\mathbf{w}} \to \underline{\mathbf{e}}_1$ (proof: $\rightsquigarrow$ group presentation)