Kernel Methods

Sargur Srihari

Topics in Kernel Methods

1. Kernel Methods *vs* Linear Models/Neural Networks
2. Stored Sample Methods
3. Kernel Functions
4. Dual Representations
5. Constructing Kernels
6. Extension to Symbolic Inputs
7. Fisher Kernel

Kernel Methods vs Linear Models/Neural Networks

- Linear parametric models for regression and classification have the form $y(\mathrm{x},\mathrm{w})$
  - During learning phase we either get a maximum likelihood estimate of w or a posterior distribution of w
  - Training data is then discarded
  - Prediction based only on vector w
- This is true of Neural networks as well
- Another class of methods use the training samples or a subset of them

Memory-Based Methods

- Training data points are used in prediction phase
- Examples of such methods
  - Parzen probability density model
    - Linear combination of kernel functions centered on each training data point
  - Nearest neighbor classification
- These are memory-based methods
- Require a metric to be defined
- Fast to train, slow to predict

Kernel Functions

- Many linear parametric models can be re-cast into equivalent dual representations where predictions are based on a kernel function evaluated at training points
- Kernel function is given by

$$k(\mathrm{x},\mathrm{x}') = \phi(\mathrm{x})^{\mathrm{T}} \phi(\mathrm{x}')$$

  - where $\phi(\mathrm{x})$ is a fixed nonlinear feature space mapping (basis function)
- Kernel is a symmetric function of its arguments

$$k(\mathrm{x},\mathrm{x}') = k(\mathrm{x}',\mathrm{x})$$

- Kernel function can be interpreted as the similarity of x and x'
- Simplest is identity mapping in feature space $\phi(\mathrm{x}) = \mathrm{x}$
  - In which case $k(\mathrm{x},\mathrm{x}') = \mathrm{x}^{\mathrm{T}}\mathrm{x}'$
  - Called Linear Kernel

Kernel Trick (or Kernel Substitution)

- Formulated as inner product allows extending well-known algorithms
  - by using the kernel trick
- Basic idea of kernel trick
  - If an input vector $\mathbf{x}$ appears only in the form of scalar products then we can replace scalar products with some other choice of kernel
- Used widely
  - in support vector machines
  - in developing non-linear variant of PCA
  - In kernel Fisher discriminant

Other Forms of Kernel Functions

- Function of difference between arguments

$$k(\mathrm{x},\mathrm{x}') = k(\mathrm{x}-\mathrm{x}')$$

  - Called *stationary* kernel since invariant to translation in space
- *Homogeneous* kernels, also known as *radial basis functions*

$$k(\mathrm{x},\mathrm{x}') = k(||\mathrm{x}-\mathrm{x}'||)$$

  - Depend only on the magnitude of the distance between arguments
- Note that the kernel function is a scalar value while x is an M-dimensional vector

For these to be valid kernel functions they should be shown to have the property $k(\mathrm{x},\mathrm{x}') = \phi(\mathrm{x})^{\mathrm{T}} \phi(\mathrm{x}')$

Dual Representation

- Linear models for regression and classification can be reformulated in terms of a dual representation
  - In which kernel function arises naturally
- Plays important role in SVMs
- Consider linear regression model
  - whose parameters are determined by minimizing *regularized sum-of-squares* error function

$$J(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathrm{w}^T\phi(\mathrm{x}_n) - t_n\right\}^2 + \frac{\lambda}{2}\mathrm{w}^T\mathrm{w}$$

where $\mathrm{w} = (\mathrm{w}_0,..,\mathrm{w}_{M\text{-}1})^{\mathrm{T}}$, $\phi = (\phi_0,..\phi_{M\text{-}1})^{\mathrm{T}}$

we have N samples $\{\mathrm{x}_1,..\mathrm{x}_N\}$

λ is the regularization coefficient

ϕ is the set of M basis functions or feature vector

  - Minimum obtained by setting gradient of $J(\mathrm{w})$ wrt w equal to zero

Solution for w as a linear combination of $\phi\,(\mathrm{x}_n)$

- By equating derivative $J(\mathrm{w})$ wrt w to zero and solving for w we get

$$\begin{aligned}
\mathrm{w} &= -\frac{1}{\lambda}\sum_{n=1}^{N}\left\{\mathrm{w}^T\phi(\mathrm{x}_n)-t_n\right\}\phi(\mathrm{x}_n) \\
&= \sum_{n=1}^{N} a_n\phi(\mathrm{x}_n) \\
&= \Phi^T \mathrm{a}
\end{aligned}$$

- Solution for w is a linear combination of vectors $\phi\,(\mathrm{x}_n)$ whose coefficients are functions of w where
  - Φ is the design matrix whose n^{th} row is given by $\phi\,(\mathrm{x}_\mathrm{n})^\mathrm{T}$

$$\Phi = \begin{bmatrix} \phi_0(x_1) & . & . & \phi_{M-1}(x_1) \\ . & & & . \\ \phi_0(x_n) & . & . & \phi_{M-1}(x_n) \\ . & & & . \\ \phi_0(x_N) & . & . & \phi_{M-1}(x_N) \end{bmatrix} \text{ is a } N \times M \text{ matrix}$$

  - Vector $\mathrm{a}=(a_1,..,a_N)^\mathrm{T}$ with the definition

$$a_n = -\frac{1}{\lambda}\left\{\mathrm{w}^T\phi(\mathrm{x}_n)-t_n\right\}$$

Transformation from $\mathbf{w}$ to $\mathbf{a}$

- Thus we have $\mathbf{w} = \Phi^T \mathbf{a}$
- Instead of working with parameter vector $\mathbf{w}$ we can reformulate least squares algorithm in terms of parameter vector $\mathbf{a}$
  - giving rise to dual representation
- We will see that although the definition of $\mathbf{a}$ still includes $\mathbf{w}$

$$a_n = -\frac{1}{\lambda}\left\{\mathbf{w}^T \phi(\mathbf{x}_n) - t_n\right\}$$

It can be eliminated by the use of the kernel function

Gram Matrix and Kernel Function

- Define the Gram matrix $K=\Phi\Phi^T$ an N ***x*** N matrix, with elements

Note: $N \, x \, M$ times $M \, x \, N$

$$K_{nm} = \phi(x_n)^T \phi(x_m) = k(x_n, x_m)$$

  - where we introduce the kernel function $k(x,x') = \phi(x)^T \phi(x')$

$$K = \begin{bmatrix} k(x_1,x_1) & . & . & k(x_1,x_N) \\ . & & & \\ . & & & \\ k(x_N,x_1) & & & k(x_N,x_{N1}) \end{bmatrix}$$

Gram Matrix Definition:
Given N vectors, it is the matrix of all inner products

  - Notes:
    - Φ is NxM and K is NxN
    - K is a matrix of similarities of pairs of samples (thus it is symmetric)

Error Function in Terms of Gram Matrix of Kernel

- Sum of squares Error Function is

$$J(\mathrm{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathrm{w}^T\phi(\mathrm{x}_n) - t_n\right\}^2 + \frac{\lambda}{2}\mathrm{w}^T\mathrm{w}$$

- Substituting $\mathrm{w} = \Phi^{\mathrm{T}}\mathrm{a}$ into $J(\mathrm{w})$ gives

$$J(\mathrm{w}) = \frac{1}{2}\mathrm{a}^T\Phi\Phi^T\Phi\Phi^T\mathrm{a} - \mathrm{a}^T\Phi\Phi^T\mathrm{t} + \frac{1}{2}\mathrm{t}^T\mathrm{t} + \frac{\lambda}{2}\mathrm{a}^T\Phi\Phi^T\mathrm{a}$$

where $\mathrm{t} = (t_1,..,t_N)^{\mathrm{T}}$

- Sum of squares error function is written in terms of Gram matrix as

$$J(\mathrm{a}) = \frac{1}{2}\mathrm{a}^T\mathrm{K}\mathrm{K}\mathrm{a} - \mathrm{a}^T K\mathrm{t} + \frac{1}{2}\mathrm{t}^T\mathrm{t} + \frac{\lambda}{2}\mathrm{a}^T K\mathrm{a}$$

- Solving for a by combining $\mathrm{w}=\Phi^{\mathrm{T}}\mathrm{a}$ and $a_n = -\frac{1}{\lambda}\left\{\mathrm{w}^T\phi(\mathrm{x}_n) - t_n\right\}$

$$\mathrm{a} = (\mathrm{K} + \lambda \mathrm{I}_N)^{-1}\mathrm{t}$$

Solution for a can be expressed as a linear combination of elements of $\phi(\mathrm{x})$ whose coefficients are entirely in terms of kernel $k(\mathrm{x},\mathrm{x}')$ from which we can recover original formulation in terms of parameters w

Prediction Function

- Prediction for new input $\mathbf{x}$
  - We can write $\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I}_N)^{-1}\mathbf{t}$ by combining $\mathbf{w} = \Phi^T \mathbf{a}$ and $a_n = -\frac{1}{\lambda}\left\{\mathbf{w}^T\phi(\mathbf{x}_n) - t_n\right\}$
  - Substituting back into linear regression model,

$$
\begin{aligned}
y(\mathbf{x}) &= \mathbf{w}^T \phi(\mathbf{x}) \\
&= \mathbf{a}^T \Phi \phi(\mathbf{x}) \\
&= \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \lambda I_N)^{-1}\mathbf{t} \;\; \text{where } \mathbf{k}(\mathbf{x}) \text{ has elements } k_n(\mathbf{x}) = k(\mathbf{x}_n, \mathbf{x})
\end{aligned}
$$

- Prediction is a linear combination of the target values from the training set.

Advantage of Dual Representation

- Solution for a is expressed entirely in terms of kernel function $k(\mathrm{x},\mathrm{x}')$
- Once we get a we can recover w as linear combination of elements of $\phi(\mathrm{x})$ using $\mathrm{w} = \Phi^t \mathrm{a}$
- In parametric formulation, solution is $\mathrm{w}_{ML} = (\Phi^T\Phi)^{-1}\Phi^T \mathrm{t}$
  - Instead of inverting an M x M matrix we are inverting an N x N matrix– an apparent disadvantage
- But, advantage of dual formulation is that we can work with kernel function $k(\mathrm{x},\mathrm{x}')$ and therefore
  - avoid working with a feature vector $\phi(\mathrm{x})$ and
  - problems associated with very high or infinite dimensionality of x

Constructing Kernels

- To exploit kernel substitution need valid kernel functions
- First Method
  - choose a feature space mapping $\phi(\mathrm{x})$ and use it to find corresponding kernel
  - One-dimensional input space

$$k(x,x') = \phi(x)^T \phi(x') = \sum_{i=1}^{M} \phi_i(x)\phi_i(x')$$

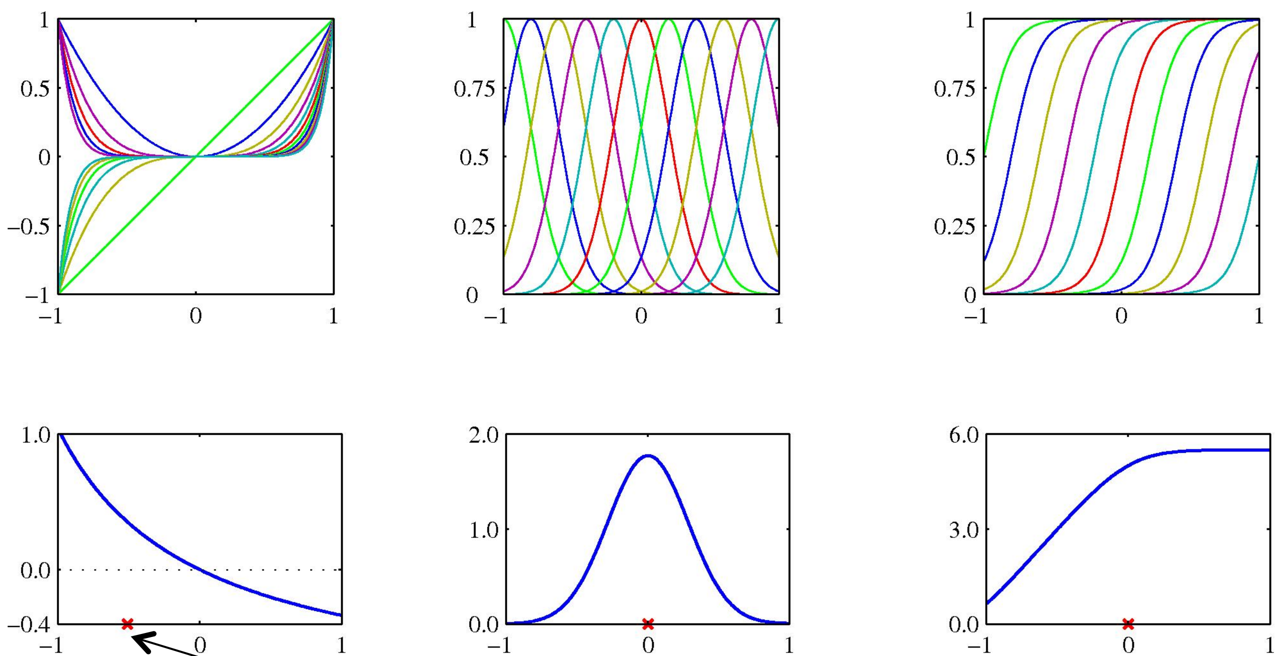
    - where $\phi(x)$ are basis functions such as polynomial
    - For each i we choose $\phi_i = x^i$

Construction of Kernel Functions from basis functions

One-dimensional input space

Polynomials Gaussian Logistic Sigmoid

Basis Functions $\phi_i(x)$



Kernel Functions $k(x,x') = \phi(x)^{\mathrm{T}}\phi(x)$

Red cross is x'

Second Method: Direct Construction of Kernels

- Function we choose has to correspond to a scalar product in some (perhaps infinite dimensional) space
- Consider kernel function $k(\mathrm{x},\mathrm{z}) = (\mathrm{x}^{\mathrm{T}}\mathrm{z})^2$
  - In two dimensional space
$$\begin{aligned} k(\mathrm{x},\mathrm{z}) &= (\mathrm{x}^T\mathrm{z})^2 = (x_1z_1 + x_2z_2)^2 \\ &= x_1^2z_1^2 + 2x_1z_1x_2z_2 + x_2^2z_2^2 \\ &= (x_1^2,\sqrt{2}x_1x_2,x_2^2)(z_1^2,\sqrt{2}z_1z_2,z_2^2)^T \\ &= \phi(\mathrm{x})^T\phi(\mathrm{z}) \end{aligned}$$
  - Feature mapping takes the form $\phi(x) = (x_1^2,\sqrt{2}x_1x_2,x_2^2)$
  - Comprises of all second order terms with a specific weighting
    - Inner product needs computing six feature values and 3 x 3 = 9 multiplications
    - Kernel function $k(x,z)$ has 2 multiplications and a squaring
- By considering $(\mathrm{x}^{\mathrm{T}}\mathrm{z}+\mathrm{c})^2$ we get constant, linear, second order terms
- By considering $(\mathrm{x}^{\mathrm{T}}\mathrm{z}+\mathrm{c})^{\mathrm{M}}$ we get all powers of x (monomials)

Testing whether a function is a valid kernel

- Without having to construct the function $\phi(\mathrm{x})$ explicitly
- Necessary and sufficient condition for a function $k(\mathrm{x},\mathrm{x}')$ to be a kernel is
  - Gram matrix K, whose elements are given by $k(\mathrm{x}_n,\mathrm{x}_m)$ is positive semi-definite for all possible choices of the set $\{\mathrm{x}_n\}$
    - Positive semi-definite is not the same thing as a matrix whose elements are non-negative
    - It means $\mathrm{z}^T K \mathrm{z} \geq 0$ for non-zero vectors z with real entries

      $$\text{i.e.,} \sum_n \sum_m K_{nm} z_n z_m \geq 0 \quad \text{for any real numbers } z_n, z_m$$
    - Mercer's theorem: any continuous, symmetric, positive semi-definite kernel function $k(x, y)$ can be expressed as a dot product in a high-dimensional space
- New kernels can be constructed from simpler kernels as building blocks

Techniques for Constructing Kernels

- Given valid kernels $k_1(\mathrm{x},\mathrm{x}')$ and $k_2(\mathrm{x},\mathrm{x}')$ the following new kernels will be valid

1. $k(\mathrm{x},\mathrm{x}') = ck_1(\mathrm{x},\mathrm{x}')$
2. $k(\mathrm{x},\mathrm{x}') = f(\mathrm{x})k_1(\mathrm{x},\mathrm{x}')f(\mathrm{x}')$
3. $k(\mathrm{x},\mathrm{x}') = q(k_1(\mathrm{x},\mathrm{x}'))$
4. $k(\mathrm{x},\mathrm{x}') = \exp(k_1(\mathrm{x},\mathrm{x}'))$
5. $k(\mathrm{x},\mathrm{x}') = k_1(\mathrm{x},\mathrm{x}') + k_2(\mathrm{x},\mathrm{x}')$
6. $k(\mathrm{x},\mathrm{x}') = k_1(\mathrm{x},\mathrm{x}')k_2(\mathrm{x},\mathrm{x}')$
7. $k(\mathrm{x},\mathrm{x}') = k_3(\phi(\mathrm{x}).\phi(\mathrm{x}'))$
8. $k(\mathrm{x},\mathrm{x}') = \mathrm{x}^T\mathrm{A}\mathrm{x}'$
9. $k(\mathrm{x},\mathrm{x}') = k_a(\mathrm{x}_a,\mathrm{x}_b') + k_b(\mathrm{x}_b,\mathrm{x}_b')$
10. $k(\mathrm{x},\mathrm{x}') = k_a(\mathrm{x}_a,\mathrm{x}_a')k_b(\mathrm{x}_b,\mathrm{x}_b')$

Where
$f(.)$ is any function

$q(.)$ is a polynomial with non-negative coefficients

$\phi(\mathrm{x})$ is a function from x to R^M
k_3 is a valid kernel in R^M
A is a symmetric positive semidefinite matrix

x_a and x_b are variables with $\mathrm{x}=(\mathrm{x}_a,\mathrm{x}_b)$
k_a and k_b are valid kernel functions

Kernels appropriate for specific applications

- Requirements for $k(\mathrm{x},\mathrm{x}')$
  - It is symmetric
  - Its Gram matrix is positive semidefinite
  - It expresses the appropriate similarity between x and x' for the intended application

Gaussian Kernel

- Commonly used kernel is

  $$k(x,x') = \exp(-||x-x'||^2/2\sigma^2)$$

- It is seen as a valid kernel by expanding the square

  $$||x-x'||^2 = x^Tx + (x')^Tx' - 2x^Tx'$$

- To give

  $$k(x,x') = \exp(-x^Tx/2\sigma^2)\exp(-x^Tx'/\sigma^2)\exp(-(x')^Tx'/2\sigma^2)$$

- From kernel construction rules 2 and 4
  - together with validity of linear kernel $k(x,x')=x^Tx'$
- Can be extended to non-Euclidean distances

  $$k(x,x') = \exp\{(-1/2\sigma^2)[\kappa(x,x')+\kappa(x',x')-2\kappa(x,x')]\}$$

Extension of Kernels to Symbolic Inputs

- Important contribution of kernel viewpoint:
  - Inputs that are symbolic rather than vectors of real numbers
- Kernel functions defined for graphs, sets, strings, text documents
- If A_1 and A_2 are two subsets of objects
  - A simple kernel is

$$k(A_1,A_2) = 2^{|A_1 \cap A_2|}$$

  - where | | indicates cardinality of set intersection
  - A valid kernel since it can be shown to correspond to an inner product in a feature space

$A=\{1,2,3,4,5\}$

$A_1=\{2,3,4,5\}$
$A_2=\{1,2,4,5\}$
$A_1 \cap A_2=\{2,4,5\}$
Hence $k(A_1,A_2)=8$

What are feature vectors $\phi(A_1)$ and $\phi(A_2)$ such that $\phi(A_1)\phi(A_2)^T=8$?

Combining Discriminative and Generative Models

- Generative models deal naturally with missing data and with HMM of varying length
- Discriminative models such as SVM have better performance
- Can use a generative model to define a kernel and use kernel in discriminative approach

Kernels based on Generative Models

- Given a generative model $p(\mathrm{x})$ we define a kernel by

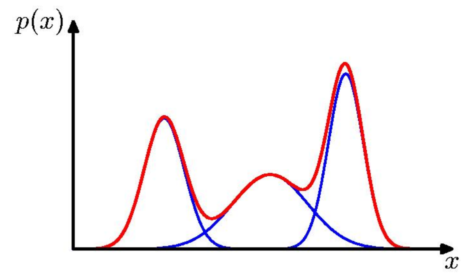
  $$k(\mathrm{x},\mathrm{x}') = p(\mathrm{x})\, p(\mathrm{x}')$$

  - A valid kernel since it is an inner product in the one-dimensional feature space defined by the mapping $p(\mathrm{x})$
- Two inputs x and x' are similar if they have high probabilities

Kernel Functions based on Mixture Densities

- Extension to sums of products of different probability distributions

$$k(\mathrm{x},\mathrm{x}') = \sum_i p(\mathrm{x}\,|\,i)\,p(\mathrm{x}'|\,i)\,p(i)$$



  - where $p(i)$ are positive weighting coefficients
  - It is a valid kernel based on two rules of kernel construction:

    $k(\mathrm{x},\mathrm{x}') = ck_1(\mathrm{x},\mathrm{x}')$ and $k(\mathrm{x},\mathrm{x}') = k_1(\mathrm{x},\mathrm{x}') + k_2(\mathrm{x},\mathrm{x}')$

- Two inputs x and x' will give a large value of k, and hence appear similar, if they have a significant probability under a range of different components
- Taking the limit to infinite sum

$$k(\mathrm{x},\mathrm{x}') = \int p(\mathrm{x}\,|\,\mathrm{z})\,p(\mathrm{x}'|\,\mathrm{z})\,p(\mathrm{z})\,d\mathrm{z}$$

  - where z is a continuous latent variable

Kernels for Sequences

- Data consists of ordered sequences of length L

$$\mathrm{X}=\{x_1,..,x_L\}$$

- Generative model for sequences is HMM
  - Hidden states $Z=\{z_1,..,z_L\}$
- Kernel Function for measuring similarity of sequences X and X' is

$$k(\mathrm{X},\mathrm{X}') = \sum_Z p(\mathrm{X}\mid \mathrm{Z})\,p(\mathrm{X}'\mid \mathrm{Z}')\,p(\mathrm{Z})$$

  - Both observed sequences are generated by same hidden sequence Z

Fisher Kernel

- Alternative technique for using generative models
  - Used in document retrieval, protein sequences, document recognition
- Consider parametric generative model $p(\mathrm{x}|\theta)$ where θ denotes vector of parameters
- Goal: find kernel that measures similarity of two vectors x and x' induced by the generative model
- Define Fisher score as gradient wrt θ

$$\mathrm{g}(\theta,\mathrm{x}) = \nabla_\theta \ln p(\mathrm{x} \mid \theta)$$

A vector of same dimensionality as θ

Fisher score is more generally the gradient of the log-likelihood

- Fisher Kernel is

$$k(\mathrm{x},\mathrm{x}') = \mathrm{g}(\theta,\mathrm{x})^T \mathrm{F}^{-1} \mathrm{g}(\theta,\mathrm{x}')$$

where F is the Fisher information matrix

$$\mathrm{F} = \mathrm{E}_\mathrm{x}\left[\mathrm{g}(\theta,\mathrm{x})\mathrm{g}(\theta,\mathrm{x})^T\right]$$

Fisher Information Matrix

- Presence of Fisher information matrix causes kernel to be invariant under non-linear parametrization of the density model $\theta \rightarrow \psi(\theta)$
- In practice, infeasible to evaluate Fisher Information Matrix. Instead use the approximation

$$\mathrm{F} \approx \frac{1}{N}\sum_{n=1}^{N} \mathrm{g}(\theta,\mathrm{x}_n)\mathrm{g}(\theta,\mathrm{x}_n)^T$$

- This is the covariance matrix of the Fisher scores
- So the Fisher kernel $k(\mathrm{x},\mathrm{x}') = \mathrm{g}(\theta,\mathrm{x})^T\mathrm{F}^{-1}\mathrm{g}(\theta,\mathrm{x}')$ corresponds to whitening of the Fisher scores
- More simply omit F and use non-invariant kernel

$$k(\mathrm{x},\mathrm{x}') = \mathrm{g}(\theta,\mathrm{x})^T\mathrm{g}(\theta,\mathrm{x}')$$

Sigmoidal Kernel

- Provides a link between SVMs and neural networks

$$k(\mathrm{x},\mathrm{x}') = \tanh(a\mathrm{x}^{\mathrm{T}}\mathrm{x}' + b)$$

  - Its Gram matrix is not positive semidefinite
  - But used in practice because it gives SVMs a superficial resembalance to neural networks
- Bayesian neural network with an appropriate prior reduces to a Gaussian process
  - Provides a deeper link between neural networks and kernel methods