Lecture 7

13. Artificial Neural Networks for Function Approximation

Motivation.

A typical control design process starts with *modeling*, which is basically the process of constructing a mathematical description (such as a set of ODE-s) for the physical system to be controlled. Note that *more accurate models are not always better*. They may require unnecessarily complex control design and analysis and more demanding computation. The key here is to model essential effects in the system dynamics in the operating range of interest. In addition, a good model should also provide some characterization of the *model uncertainties* – the so-called “unknown unknowns”, which can be used for robust design, adaptive design, or merely simulation and system testing, (such as Monte Carlo runs). Model uncertainties are the differences between the model and the real physical process. Uncertainties in parameters are called *parametric*, while the others are called *non-parametric* uncertainties.

Example 13.1

For the model of a controlled mass $m\ddot{x} = u$ the uncertainty in m is parametric, while the neglected motor dynamics, measurement noise, sensor dynamics represent the non-parametric uncertainties.

Example 13.2

Consider the scalar model with uncertain dynamics $\dot{x} = f(x) + u$, where $f(x)$ is not known. Suppose that

$$f(x) = \sum_{i=1}^{N} \theta_i \varphi_i(x) + \varepsilon(x) = \underbrace{\theta^T \Phi(x)}_{\text{Parametric Uncertainty}} + \underbrace{\varepsilon(x)}_{\text{Non-Parametric}}$$

that is suppose that the unknown function $f(x)$ can be approximated by a linear combination of *known basis functions* $\varphi_i(x)$ and *unknown constant parameters* θ_i. The approximation error $\varepsilon(x)$ is the non-parametric uncertainty, while the unknown constant parameters θ represent the parametric uncertainty in the system dynamics. In order to characterize the latter, one needs to be able to find a good set of basis functions $\Phi(x)$ such that the approximation error $\varepsilon(x)$ becomes small on a compact x – domain. Polynomials, Fourier series expansions, splines and *feedforward neural networks* can be used to approximate functions on compact domains.

In what follows, we show how to *adapt to parametric uncertainties*, while maintain *robustness in the presence of non-parametric uncertainties*.

Definition 13.1

Artificial Feedforward Neural Networks are multi-input-multi-output systems composed of many inter-connected nonlinear processing elements (neurons) operating in parallel.

Figures 13.1 and 13.2 show sketches of two feedforward NN-s.

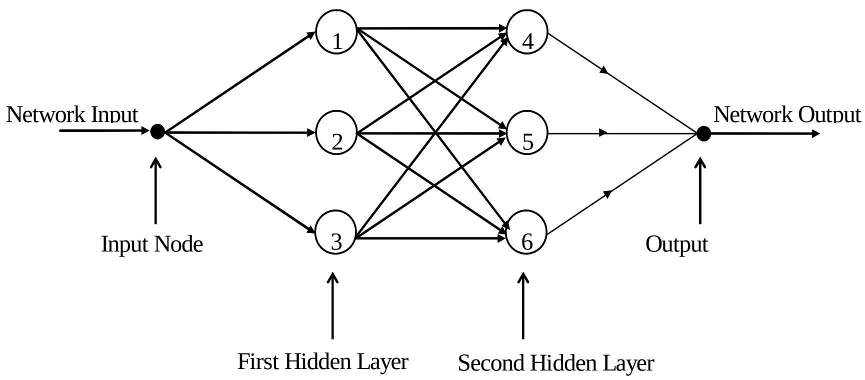

Figure 13.1: Feedforward Neural Network with 2 hidden layers and 6 neurons

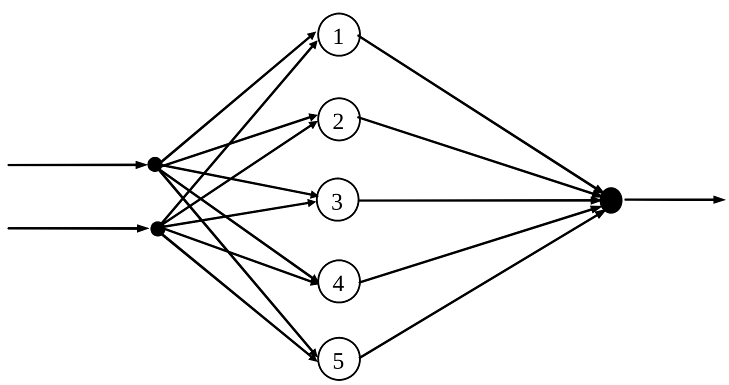

Figure 13.2: Feedforward Neural Network with 1 hidden layer and 5 neurons

As seen from these two examples, an artificial feedforward neural network consists of neurons and their connections. A block-diagram of a neuron is shown below.

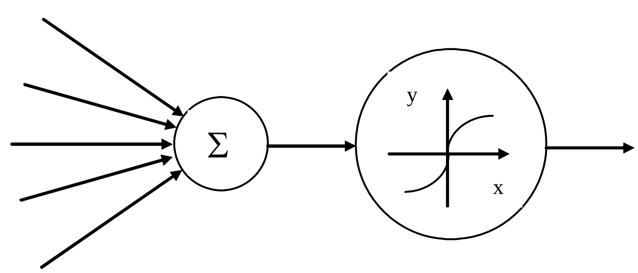

Figure 13.3: Artificial Neuron Block-Diagram

Neurons, the basic processing elements of NN-s, have two main components:

- a weighted summer
- a *nonlinear* activation function

The activation functions of interest are *Radial Basis Functions* or ridge functions, (often called the *sigmoids*).

Definition 13.2

A Radial Basis Function (RBF) is defined as a Gaussian:

$$\varphi(x, x_c) = e^{-(x-x_c)^T W (x-x_c)} = e^{-\|x-x_c\|_W^2} \tag{13.1}$$

In (13.1), $x \in R^n$ is the input, $x_c \in R^n$ is the center, and $W = W^T > 0$ is a positive-definite symmetric matrix of weights. Most often we will write $\varphi(x, x_c) = \varphi_i(x)$ to abbreviate and denote an RBF which is centered at the i^{th} center.

Remark 13.1

Other definitions of an RBF are available. Often in the literature, an RBF is defined as $\phi = \phi(\|x - x_c\|_W)$, where $\|x\|_W = \sqrt{x^T W x}$ denotes the weighted Euclidean norm of a vector x. In addition, it is required that $\phi(x)$ be integrable on R^n, and $\int_{R^n} \phi(x) dx \neq 0$. Basically, this particular RBF depends only on the weighted distance $r = \|x - x_c\|_W$ between its current input x and the center x_c. The Gaussian RBF in (13.1) is an example of this type of activation function. Others include:

- Multiquadrics

$$\varphi(r) = \sqrt{(r^2 + c^2)}, \quad c > 0$$

- Inverse multiquadrics

$$\varphi(r) = \frac{1}{\sqrt{(r^2 + c^2)}}, \quad c > 0$$

Micchelli's Theorem

Let $\varphi = \varphi(r)$ be the Gaussian, the multiquadrics, or the inverse multiquadrics function. Let $\{x_i\}_{i=1}^N$ be a set of distinct points in R^n. Then the $(N \times N)$– interpolation matrix Φ, whose $(i, j)^{th}$ element is $\varphi_{ij} = \varphi(\|x_i - x_j\|)$, is nonsingular.

Remark 13.2

There is a large class of RBF-s that is covered by Micchelli's theorem. The theorem provides theoretical basis for RBF based function approximation problems. In other words, using an RBF $\varphi = \varphi(r)$ and a finite set of N points $\{x_i\}_{i=1}^N$ in R^n, it is always possible to approximate a large class of functions $f(x)$ with $\hat{f}(x)=\sum_{i}^{N}\theta_i\,\varphi(x-x_i)$ such that $f(x_i)=\hat{f}(x_i)$, for all $\{x_i\}_{i=1}^{N}$.

Definition 13.3

A ridge function / sigmoid is a *nonlinear* function of the form:

$$\sigma=\sigma\left(w^T x+b\right) \tag{13.2}$$

where $w\in R^n$ denotes the vector of weights, b is a scalar called the threshold, and $\sigma(s)$ is a nonlinear function (not necessarily continuous), defined in R with the following properties:

$$\begin{cases}\lim\limits_{s\to-\infty}\sigma(s)<\infty\\ \lim\limits_{s\to\infty}\sigma(s)<\infty\end{cases} \tag{13.3}$$

The two most common examples of a ridge function are:

- The logistic sigmoid

$$\sigma(s)=\frac{1}{1+e^{-s}} \tag{13.4}$$

- The hyperbolic tangent

$$\sigma(s)=\frac{1-e^{-s}}{1+e^{-s}} \tag{13.5}$$

A feedforward NN with N neurons in its hidden layer is shown in Figure 13.4.

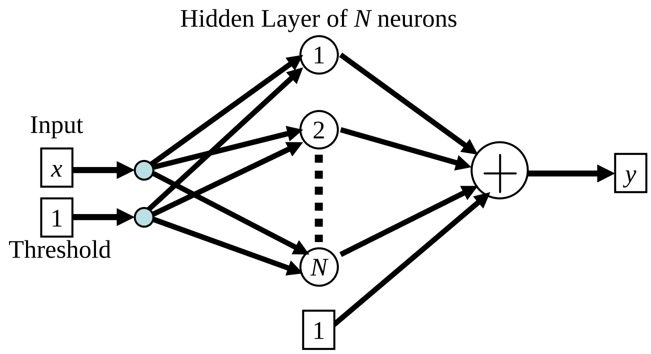

Figure 13.4: Single-hidden-layer feedforward NN with N neurons

Formally speaking, a feedforward NN maps R^n into R^m, that is:

$$y=NN(x),\quad x\in R^n,\quad y\in R^m \tag{13.6}$$

Definition 13.4

A *sigmoidal feedforward NN* is:

$$NN(x) = W^T \vec{\sigma}\left(V^T x + \theta\right) + b \tag{13.7}$$

where $W \in R^{N\times m}$ is the matrix of the *outer-layer weights*,

$$\vec{\sigma}(x) = \left(\sigma\left(V_1^T x + \theta_1\right) \quad \ldots \quad \sigma\left(V_N^T x + \theta_N\right)\right)^T \in R^N$$

is the vector of N sigmoids, $V \in R^{n\times N}$ is the matrix of the *inner-layer synaptic weights* with its i^{th} column denoted by $V_i \in R^n$, $\theta \in R^N$ is the *vector of thresholds*, and $b \in R^m$ denotes the NN *bias* vector.

Definition 13.5

A feedforward *RBF NN* is:

$$NN(x) = \theta^T \begin{pmatrix} \varphi\left(\left\|x - C_1\right\|_{W_1}\right) \\ \vdots \\ \varphi\left(\left\|x - C_N\right\|_{W_N}\right) \end{pmatrix} + b = \underbrace{\left(\theta^T \quad b\right)}_{\Theta^T} \underbrace{\begin{pmatrix} \varphi_1(x) \\ \vdots \\ \varphi_N(x) \\ 1 \end{pmatrix}}_{\Phi(x)} = \Theta^T \, \Phi(x) \tag{13.8}$$

where $\Theta = \left(\theta^T \quad b\right)^T \in R^{(N+1)\times m}$ is the *vector of weights*, $C_i \in R^n$ is the *center* of the i^{th} receptive field, $W_i = W_i^T > 0$ is the norm weighting matrix, $b \in R^m$ is the NN *bias*, and $\Phi(x) = \left(\varphi_1(x) \quad \ldots \quad \varphi_N(x) \quad 1\right)^T \in R^{N+1}$ is the so-called *regressor vector*, whose components are the basis activation functions $\varphi_i(x) = \varphi\left(\left\|x - C_i\right\|_{W_i}\right)$ and the unity function.

Remark 13.3

Often in practical applications, the symmetric positive-definite matrix W in (13.8) is chosen to be diagonal and in the form:

$$W_i = \frac{1}{2\sigma_i^2}, \quad \left(i = 1, \ldots, N\right)$$

where σ_i represents the *width* of the i^{th} Gaussian function, that is:

$$\varphi_i(x) = e^{-\frac{\left\|x - C_i\right\|^2}{2\sigma_i^2}}$$

becomes the i^{th} component of the regressor vector $\Phi(x)$ in (13.8). Most often, the components of the regressor are constructed using the *isotropic Gaussian* function:

$$\varphi_i(x) = e^{-\left(\frac{N}{d_{\max}^2}\right)\left\|x - C_i\right\|^2}$$

whose standard deviation (i.e., width) σ is fixed according to the spread of the centers C_i, N is the number of centers, and $d_{\max}$ is the maximum distance between the chosen centers. In this case, the standard deviation σ of all the isotropic Gaussian RBF components is fixed at

$$\sigma = \frac{d_{\max}}{\sqrt{2N}}$$

This formula ensures that the individual RBF-s are not too peaked or too flat. Both of these two extreme conditions should be avoided.

Feedforward NN-s have been shown to be capable of approximating generic classes of functions, including continuous and integrable ones, on a compact domain and to within any tolerance. This property of feedforward NN-s is often referred to as the *Universal Approximation* property, while the NN-s themselves are often called the *universal approximators*. Two related theorems are given below.

Universal Approximation Theorem for Sigmoidal NN-s, (G. Cybenko, 1989)

Any continuous function $f(x): R^n \to R$ can be uniformly approximated by a single-hidden-layer NN with a bounded monotone-increasing continuous activation function and on a compact domain $X \subset R^n$, that is:

$$\forall \varepsilon > 0 \quad \exists N, W, b, V, \theta \quad \forall x \in X \subset R^n \quad \left\| \underbrace{W^T \vec{\sigma}\left(V^T x + \theta\right) + b}_{NN(x)} - f(x) \right\|_\infty \le \varepsilon \quad (13.9)$$

Remark 13.4

The universal approximation theorem extends to the class of L_1 functions on a compact domain. In that case, it is assumed that the activation function is a bounded measurable sigmoid and the approximation is understood in terms of the L_1 norm.

Rate of Approximation Theorem for Sigmoidal NN-s, (12. Barron, 1993)

Consider a class of functions $f(x)$ on R^n for which there is a Fourier representation of the form

$$f(x) = \int_{R^n} e^{i\omega x}\, \tilde{f}(\omega)\, d\omega$$

for some complex-valued function $\tilde{f}(\omega)$ for which $\omega \tilde{f}(\omega)$ is integrable, and define

$$C_f = \int_{R^n} \|\omega\| \left\| \tilde{f}(\omega) \right\| d\omega < \infty$$

Then for every function $f(x)$ with C_f finite, and every $N \ge 1$, there exists a sigmoidal NN of the form (13.7), such that

$$\| f(x) - NN(x) \|_{L_2}^2 = \int_{\|x\| \le r} \left(f(x) - NN(x) \right)^2 dx \le \frac{\left(2 r C_f\right)^2}{N}$$

Remark 13.5

Functions with C_f finite are continuously differentiable on R^d. Moreover, the NN approximation error is measured by the integrated squared error, (L_2 – norm), on the ball of radius r.

Universal Approximation Theorem for RBF NN-s, (Park and Sandberg, 1991)

Let $\varphi(x): R^n \to R$ be an integrable bounded continuous function and assume that

$$\int_{R^n} \varphi(x)\,dx \neq 0$$

Then for any continuous function $f(x)$ and any $\varepsilon > 0$ there is an RBF NN with N neurons, a set of centers $\{C_i\}_{i=1}^N$, and a common width $\sigma > 0$

$$\hat{f}(x) = \sum_{i=1}^{N} \theta_i \underbrace{\varphi\left(\frac{x - C_i}{\sigma}\right)}_{\varphi_i(x)} = \Theta^T \Phi(x)$$

such that

$$\left\| f(x) - NN(x) \right\|_{L_2}^2 = \int_{\|x\| \le r} \left(f(x) - NN(x) \right)^2 dx \le \varepsilon = \mathrm{O}\left(N^{-\frac{1}{n}} \right)$$

Comparison of sigmoidal and RBF NN-s

- RBF and sigmoidal NN-s are universal approximators.
- RBF NN depends the Euclidean distances between the input vector x and the centers C_i. Meanwhile, sigmoidal NN-s depend on the inner product of the input vector x with the synaptic weight vectors V_i and biased by θ.
- Sigmoidal NN-s provide $\mathrm{O}\left(N^{-\frac{1}{2}} \right)$ rate of approximation which does not explicitly depend on the dimension of x. On the other hand, the rate of approximation for the RBF NN-s is of order $\mathrm{O}\left(N^{-\frac{1}{2n}} \right)$ and, consequently it decreases exponentially as the dimension of the input vector x increases. This phenomenon is called the Curse of Dimensionality, (due to R. Bellman).
- An RBF has local support while a sigmoid does not. The local support implies learning and adaptation ability of RBF NN-s. Sigmoidal NN-s adapt but don't learn.

With specific reference to NN-s in control, it is their ability to represent nonlinear mappings, and hence to model nonlinear systems, which is the feature to be most readily exploited in the synthesis of nonlinear controllers.

Reading material / References:

1) F. Scarselli, C. Tsoi, “*Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results*”, Neural Networks, vol. 11, No. 1, pp. 15-37, 1998.
2) K.J. Hunt, D. Sbarbaro, R. Zbikowski, P.J. Gawthrop, “*Neural Networks for Control Systems – A Survey*”, Automatica, vol. 28, No. 6., pp. 1083-1112, 1992.
3) G. Cybenko, “*Approximation by superposition of a sigmoidal function*”, Math. Control Signals Systems, vol. 2, pp. 303-314, 19813.
4) J. Park, I.W. Sandberg, “*Universal approximation using radial-basis-function networks*”, Neural Computation, vol. 3, no. 2, pp. 246-257.
5) C. Micchelli, “*Interpolation of scattered data: Distance matrices and conditionally positive definite functions*”, Constructive Approximation, vol.2, pp. 11-12, 1986.