Neuro Adaptive Control

Suppose we had an unknown nonlinearity (matched) in the plant

$$\dot{x} = Ax + B\Lambda\left(u + f(x)\right) \quad , \quad x(0) = x_0 \quad , \quad x \in \mathbb{R}^n$$

How does one handle the unstructured case, e.g., nothing is known about f?

Well, one can attempt to approximate it with a structured nonlinear function. In this case, that means we try to find a minimal error parametrization for the nonlinearity,

$$f(x) = \underbrace{\alpha^T \Phi(x)}_{\text{parametric uncertainty}} + \underbrace{\varepsilon(x)}_{\text{non-parametric uncertainty}}$$

(approximation error of parametrization)

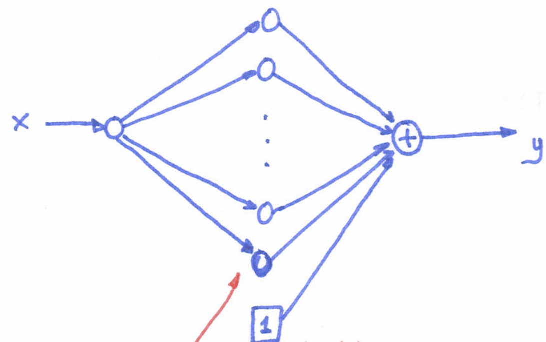
Approximating $f(x)$ well means finding a good set of basis functions $\Phi(x)$ minimizing the error on a compact subset of the domain.

- subset should be of interest and realistically encompass the achievable state under the desired controller.

- Many methods exist: polynomials, splines, Fourier series, feedforward neural networks, ...

Of course, the method of choice for this section is feedforward neural networks. The network itself will be composed of radial basis functions.

· A neural network is one way of visualizing a specific class of mathematical constructs. It's a way of combining nonlinear and linear operations to represent an input-output system.



output bias

these nodes have some sort of nonlinear processing going on.

* pictured is $f_{NN}: \mathbb{R}^n \rightarrow \mathbb{R}$

Many such NN's used to achieve $f_{NN}: \mathbb{R}^n \rightarrow \underline{\mathbb{R}^m}$
vector output.

The function $y = f(x)$ is approximated by the neural network $y \approx f_{NN}(x)$.

In our case, radial basis functions are the nonlinear processing elements, AKA the activation functions. A radial basis function has the form

$$\varphi(x\,;\, x_c, \Sigma) = e^{-\|x - x_c\|_\Sigma^2}$$

· RECALL: $\|\xi\|_\Sigma = \xi^T \Sigma\, \xi$

symmetric, positive definite.

Since it is the magnitude of the norm that really matters, an alternative definition is sometimes used,

$$\varphi(r) = e^{-r^2}$$

then we'd use $r = \|x - x_c\|_\Sigma$ as the argument.

Theorem (Michelli's Theorem). Let $\varphi = \varphi(r)$ be the Gaussian radial basis function. Let $\{x_i\}_1^N$ be a set of N distinct points in $\mathbb{R}^n$. Then the $N \times N$ interpolation matrix Φ, where $\varphi_{ij} = \varphi(\|x_i - x_j\|)$, is nonsingular

- $\Phi = [\varphi_{ij}]_{1\ldots N, 1\ldots N}$ in the theorem statement.

- The theorem is relevent to the discussion because it has utility in solving for the function approximation. How?

Take an RBF, φ, and a collection of points in $\mathbb{R}^n$, $\{x_i\}_1^N$. The function $f: \mathbb{R}^n \to \mathbb{R}$ can be approximated using

$$\hat{f}(x) = \sum \alpha_i \varphi(x - x_i)$$

such that $f(x_i) = \hat{f}(x_i)$ for all x_i, $i = 1 \ldots N$.

achieving these constraints requires the inverse to the interpolation matrix (i.e., when solving for the α_i).

- As part of a neural network (feed-forward), the approximation is:

$$\hat{f}_{NN}(x) = \alpha^T \begin{Bmatrix} \varphi(\|x - x_1\|_{\Sigma_1}) \\ \vdots \\ \varphi(\|x - x_N\|_{\Sigma_N}) \end{Bmatrix} + \underbrace{\beta}_{\text{bias (vector)}} = [\alpha^T \ \beta] \begin{Bmatrix} \varphi(\|x - x_1\|_{\Sigma_1}) \\ \vdots \\ \varphi(\|x - x_N\|_{\Sigma_N}) \\ 1 \end{Bmatrix}$$

$$= W^T \Phi(x)$$

$W = \begin{bmatrix} \alpha \\ \beta^T \end{bmatrix}$

Practical simplifications:

- choose $\Sigma_i = \frac{1}{2\sigma_i^2}$ $\Rightarrow$ $\varphi(x; x_i, \sigma_i) = e^{-\frac{\|x - x_i\|^2}{2\sigma_i^2}}$

- usually can pick all σ_i equal

$$\varphi_i(x) \equiv \varphi(x; x_i, \sigma) = e^{-\frac{\|x - x_i\|^2}{2\sigma^2}}$$

One such option is to uniformly distribute the data points over the domain and then choose, $\sigma_i = \sigma = \frac{d^2}{2N}$, where d is the minimal distance between centers.

Now, how well does this strategy work? (as far as approximations go.)

Theorem. (Universal approximation theorem for RBF NN's).

Let $\varphi(\cdot): \mathbb{R}^n \to \mathbb{R}^n$ be an integrable, bounded, continuous function and assume that

$$\int_{\mathbb{R}} \varphi(x)dx \neq 0.$$

Then, for any continuous function $f: \mathbb{R}^n \to \mathbb{R}$ and any $\varepsilon > 0$ there is a RBF NN with N neurons, a set of centers $\{x_i\}_1^N$, and a common width $\sigma > 0$, satisfying

$$f_{NN}(x) = \sum_{i=1}^{N} \alpha_i \varphi\left(\frac{x - x_i}{\sigma}\right) = \alpha^T \Phi(x)$$

such that

$$\|f(x) - f_{NN}(x)\|_2^2 = \int_{\|x\| \leq r} [f(x) - f_{NN}(x)]^2 dx \leq \varepsilon = \Theta(N^{-1/n})$$

for some $r < \infty$.

dimension of domain

- so, approximation is of order $\theta(N^{-\frac{1}{2n}})$.

\# of sample points

- f_{NN} defined everywhere, but has the specified error only on a compact domain (outside of which, there are no promises).

- Although described as being a feedforward neural network, one need not necessarily interpret it in that manner only. This type of approximation can be found in many other areas without reference to neural nets.

- There are other "neural net" architectures/designs possible and alternative functions to radial basis functions.

When used within the context of an adaptive control system, the use of RBF NN's will add a few more checks and steps to the procedure.

$$\dot{x} = Ax + B\Lambda(u + f(x))$$

In our control, we will use an approximation to $f(x)$, hence we'll define/rewrite it to be

$$f(x) = f_{NN}(x) + \varepsilon(x) = W^T\Phi(x) + \varepsilon(x)$$

↳

$$\dot{x} = Ax + B\Lambda(u + W^T\Phi(x) + \varepsilon(x))$$

- then in the control u, there will be a $\hat{W}^T(t)\Phi(x)$ term.

  The approximation error cannot be cancelled.

- Furthermore, there will be an Approximation Assumption,
  - The number of RBF NN nodes N, the true weights W^*, and the widths $\vec{\sigma}$ are defined such that the RBF NN approximates the nonlinearity to a given tolerance

$$\|\varepsilon(x)\| = \|f(x) - (W^*)^T\Phi(x)\| \leq \varepsilon_{tol} \quad \forall x \in D_{NN} \subset \mathbb{R}^n$$

compact domain of best approximation.

- The rest follows as for the standard system with state-bounded disturbances. (Projection, deadzone, σ/e-mod, etc.)

necessary to keep adaptive states bounded.