

Research Summary
Of
Peter R. Conwell, PhD

We work in an area broadly classified as *Machine Learning*. We study ways to help computers extract data from the environment, synthesize models from that data, use these models to interact intelligently, and modify their behavior accordingly. In the early years of artificial intelligence, research focused on getting machines to mimic human performance on tasks stereotypically considered to require intelligence, like playing chess or doing calculus. Because pattern recognition was something humans (and some animals) did easily, without thought, we dismissed it. We now know that pattern recognition is essential to most intelligent tasks—like recognizing hand written characters or spoken words. Marvin Minsky, Massachusetts Institute of Technology A.I. guru, once said, “*In general, we are least aware of what our minds do best.*” It’s true. Getting machines to recognize patterns with anywhere near human performance is remarkably hard. Pattern recognition leverages all of machine learning. To recognize patterns well, computers must extract data from the environment; form internal representations, and adapt.

The systems we study are *neural* models—parallel distributed processing systems that are crude representations of biological nervous systems. We call them neural because they consist of neuron-like processing elements connected together with structures loosely associated with synapses. These *neural networks* interact with the environment. They take in data, compress it, and distribute chunks of information to the various processing elements. Since the network is highly interconnected, these, nodes can combine disparate chunks and thereby synthesize novel outputs. The direction of information flow is important. If signals proceed in one direction from the environment to input neurons, from input neurons to other internal neurons, and from these eventually to output neurons, we call the network *feed forward*. These networks are inherently static filters. Time has little meaning in these systems.

However, there is another class of networks where time is essential. These are *recurrent* networks. If a network has signals circulating back to some intermediate point, we call them recurrent. Recurrent networks can have highly varied behavior. They constitute a *non-linear dynamic* system. *Linear* dynamic systems have limited behavior. Signals oscillate, grow without bound, or decay to a static value. Recurrent networks can do all that and more. Neurons can change state suddenly, like switches. Signals can fluctuate wildly; they can even be chaotic. They can adapt and learn—without changing connection weights. Circulating signals represent a kind of memory. What’s more, these networks have been shown to be *Turning equivalent*. That is, they can emulate any other computer.

Be they feedforward or recurrent, we train our models by adjusting the strength (or weight) of the connections between the neuron-like processing elements. Over the last twenty years, statistical learning theory has evolved a rich body of mathematics that gives a strong foundation to the training algorithms for feedforward networks. From this base, there has emerged a host of machine learning algorithms. Some of these algorithms, like *radial basis functions* (RBFs) have long and rich history. We use RBFs to do function interpolation—just another name for model building. We approximate a given function with combinations of RBFs. This is not unlike approximating periodic functions with combinations of sines and cosines. RBFs are similar to kernel methods used in estimating probability density functions. RBF’s take on a slightly different shade when we view them as *support vector machines* (SVMs). (The term *machine* here is a misnomer. It refers to software, not hardware. In particular, it refers to how the network is trained) SVMs utilize a simple, but counter-intuitive, idea. They map the incoming data patterns into a high dimensional space. In this representation, because of the high dimensionality, patterns spread out, and are relatively easy to separate. In fact one can often define a no man’s land between different patterns. Two parallel hyperplanes, we call *support vectors*, bound this region. The training algorithm attempts to maximize the distance between these hyperplanes. Both RBFs and SVMs are *non-linear* techniques because they attempt to model incoming data with functions that are more complex than simply adding the incoming information. Such a non-linear transformation enables them to model more complex data than linear techniques. Some of training algorithms for feedforward neural networks can be adapted to recurrent networks. However, because of their dynamic behavior, training recurrent networks is hard.

We have focused our research to medical applications of feedforward networks and the evolutionary training of recurrent networks. In evolutionary training, we construct ensembles of recurrent neural

networks, alter them via mutation, and reproduce the successful ones. Sometimes we combine successful systems and derive a *child* that may have characteristics of its parents. Interestingly, we apply *statistical physics*. Occasionally, as we evolve them, the mean performance of the ensemble changes radically. The networks undergo a *change of phase*. There is precedent. Physicists, computer scientists, and mathematicians have demonstrated similar changes in a host of complex systems. Physicists see phase transitions in auto traffic models. Mathematicians see them in abstract systems called graphs. Computer scientists see them in ensembles of computer programs. There are broader implications here. These results are telling us about complex systems in general.

Our medical applications utilize feedforward networks to make better predictions of medical outcomes. Working with the Alex Goldfarb MD and John Hurdle MD, PhD, of the Veteran's Hospital of Salt Lake, and the University of Utah, we are trying to predict how long a person will live after a kidney transplant. If the prognosis is poor, dialysis may offer a better outcome. This frees up a scarce kidney for another patient that might have a better outcome. Traditionally, based on hundreds of parameters like age, gender, race, etc., medical researchers use simple logistic regression to predict life expectancy post transplant. We use radial basis function networks and support vector machines to make better longevity predictions. The national kidney database consists of statistical data on more than ninety thousand kidney recipients. We use this database to train our computers. This is a form of *data mining*. We mine this huge database to construct statistical models of parameters that conspire to affect a transplant outcome. Typically, this training takes weeks running on a number of Pentium based machines. We have five such machines each one running a neural model with different parameters. Just as nature required millennia to evolve our nervous system, these neural models require a good deal of fine-tuning.

There are three of us: Mayo Miller (Jr., physics and mathematics), Cory Kearns (Jr., computer science, physics minor), and me, Dr. Peter R. Conwell (Assistant Professor, physics). All of us traveled to the National Conference of Undergraduate Research (NCUR) in April in Virginia. Mayo and Cory presented papers of last summer's research funded by Gore summer research grants. Here are the abstracts of their papers:

Predicting the Outcome of Kidney Transplants using Artificial Neural Networks

Mayo Miller
Department of Physics
Westminster College
1840 South 1300 East
Salt Lake City, UT 84105. USA

Faculty Advisor: Dr. Peter Conwell

Abstract

In 1988, the United States Renal Data System (USRDS) was created by the National Institute of Diabetes and Digestive and Kidney Diseases in conjunction with the Health Care Financing Administration. These organizations constructed a data set that contained information on every patient that received a kidney transplant in the United States beginning at January 1st, 1990. There was a shortage of kidneys available for transplant; in 2001, only 27% of those patients waiting for a kidney received a transplant. Thus, it was important to identify those patients who are likely to reject a transplant. Physicians model their patients with logistic regression to predict the success or failure of future transplants. These models compare a patient's features, like age, hypertension, diabetes, with those mined from the data base. However, logistic regression lacks the ability to model complex non-linear interactions among the features that make up a patients model. Artificial Neural Networks (ANN) and Radial Basis Functions (RBF) are data mining techniques with the capacity of modeling complex, non-linear (and linear) data sets. However, they are not widely used to mine medical databases. RBFs are linear combinations of radially symmetric nonlinear basis functions. Our principle hypothesis was that using non-linear modeling techniques, such as ANNs and RBFs, we would obtain better predictions than logistic regression. Trained RBF networks with approximately 50,000

nodes were used to predict the survival of kidney transplants. In fact, RBFs outperformed logistic regression in terms of predicting survival.

Data Mining The National Kidney Database With Self-Organizing Feature Maps

Cory Kearns
Department of Computer Science
Westminster College
1840 South 1300 East
Salt Lake City, UT 84105. USA

Faculty Advisor: Dr. Peter Conwell

Abstract

Here **Kohonen Self-Organizing Feature Maps (KSOFM)** are used for data mining on the **National Kidney Transplant Database**. We characterize and classify patients into a group that is likely to do well with a kidney transplant and another that is likely to reject a transplant or otherwise do well with dialysis only. The database consists of 92,844 records of patients with end-stage renal disease (**ESRD**). Having an effective method that can identify a patient unlikely to benefit from a kidney transplant provides the choice for the patient to remain on dialysis and there by free up a kidney. The parameters that describe and govern the characteristics of the two sets interact in a complex non-linear manner. **KSOFMs** offer a way to capture and characterize these interactions. Two separate **KSOFMs** were trained: one using inputs with graft survivals and the other with graft failures. The trained maps were combined to form one large map. The combined map constituted a new and novel classification technique. The performance of the classifier was tested on the kidney database. The performance was comparable to that of logistic regression, the gold standard used in medical research.

We are always looking for other mathematics, physics, and computer science students to join us. Send me an email and introduce yourself, pconwell@westminstercollege.edu.