PAMUKKALE ÜNİVERSİTESİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
Recurrent Neural Network

Raşit EVDÜZEN

December 25, 2019

Abstract

Yapay öğrenme alanındaki çalışmalar ve algoritmalar veri (data) odaklı olup, veri tiplerine göre değişiklik göstermektedir.El yazısı tanıma, konuşma tanıma, sıralı işlemler gibi verilerle uğraşıldığı zaman klasik yapay sinir ağları çok etkili olmamaktadır.Veriler arasına zaman bilgisi eklendiği zaman RNN veya LSTM yapıları kullanılmaktadır.RNN yapılarında bir geri besleme yapısı bulunmaktadır, bu durum geri yayılım algoritmasını zorlaştırmaktadır.Figure1 bazı kullanılan yapay sinir ağları modelleri görünmektedir.

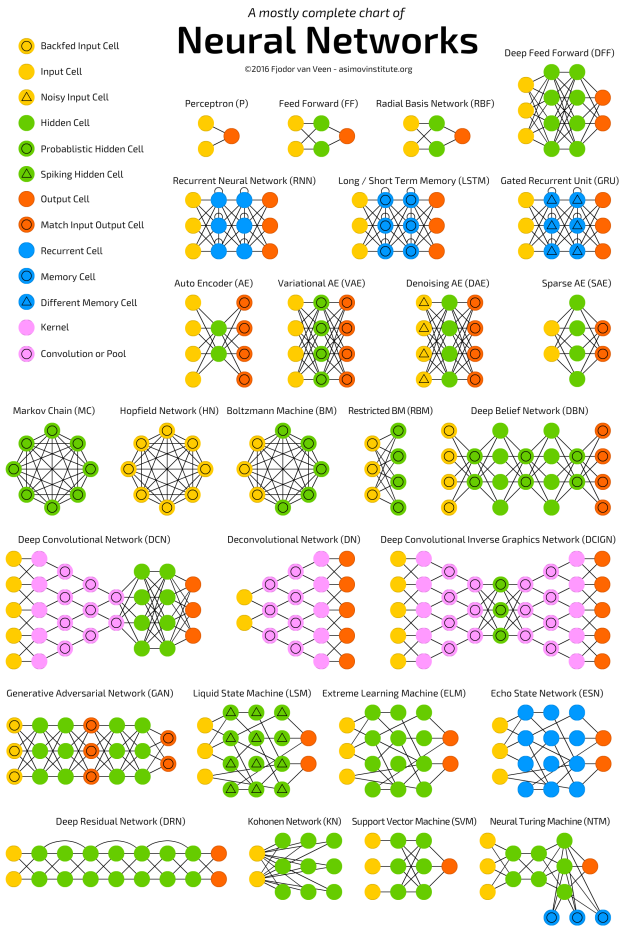

Figure 1: Yapay Sinir Ağları Topolojisi

Sıralı Veri (Sequential data)

Sıralı veriler; doğal dil işleme,zaman serisi analizi, konuşma tanıma ve genetik bilimi gibi çok çeşitli alanlarda bulunmaktadır.Genel olarak RNN'ler zaman serilerini ve sıralı olan verileri modellemek için kullanılırlar.Mesela girdi olarak 2 3 1 2 3 1 2 3 1 2 3 gibi veriler olabilir, verilerde arka arkaya gelen her 3 karakter birbirini tekrar etmektedir ve hedefimiz 4. karakterin ne olduğunu bulmak olacaktır.Girdilerimiz bir harf dizisi de olabilir örnek olarak "hello" kelimesi "h e l l o" olarak ifade edilmektedir ve aşağıda RNN yapısı üzerinde gösterilmiştir.

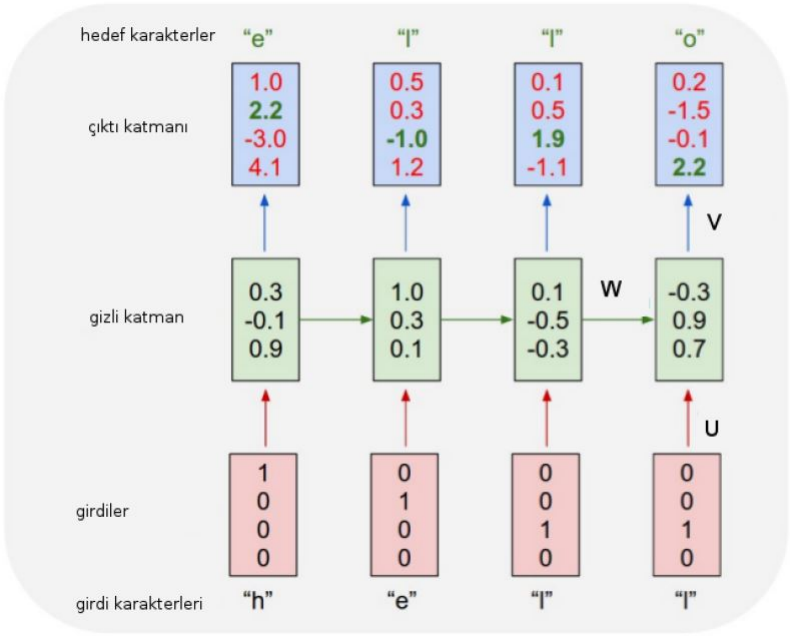

Figure 2: RNN Yapısı

Klasik ileri beslemeli yapay sinir ağı (Feed Forward Neural Network), N boyutlu veriyi alıp tüm veriyi aynı anda işlemektedir.Böylece 1. epoch sonuna gelindiği bir hata vektörü oluşturulmaktadır.Oluşturulan bu hata vektörü sayesinde back propagation algoritması ile sinir ağının parametreleri güncellenmektedir.Fakat bir RNN yapısı böyle çalışmamaktadır.

Recurrent Neural Network Yapısı

Elimizde gözlem verileri $x = [x_1, x_2, ..., x_T]^{'}$ ve bunlarla ilişkili olarak $y = [y_1, y_2, ..., y_T]^{'}$ olduğunu farz edelim.Amacımız $f : x \rightarrow y$ f fonksiyonunu (Nonlinear mapping) bulmaktır.

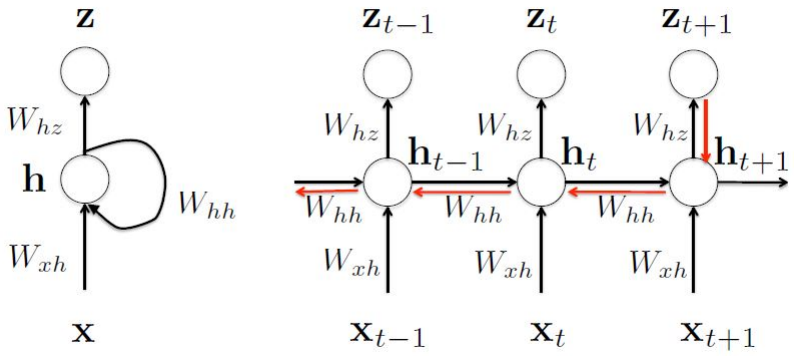

Figure 3: Soldaki grafik RNN recursive yapısını, sağdaki grafik RNN zamana bağlı olarak genişletilmiş hali göstermektedir

RNN yapısı bir dinamik sistemdir, h_t gizli durum(state) oluşmaktadır bu katman t anındaki verilere x_t tümüyle bağlı değildir. h_t durumu, geçmiş durum h_{t-1}'ye de bağlıdır.Matematiksel olarak;

$$h_t = f(h_{t-1}, x_t)$$

f :(nonlinear mapping-tanh) yukarıdaki yapıdan görüldüğü gibi RNN gizli durumları uzun vadeli bilgi saklamak için kullanmaktadır.Bu da RNN yapısına bir bellek özelliği eklemektedir.Figür3 deki Rnn modelinin matematiksel ifadesi;

$$h_t = tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

$$z_t = softmax(W_{hz}h_t + b_z)$$

RNN isminde tekrarlama (recursive) adı W_{hh} ağırlık larının her zaman adımında aynı olmasından gelmektedir.Paylaşılan ağırlıklar ile modelin iyi genelleme yapması sağlanabilir.W_{hh} ağırlık matrisi geçmişe ne derece bağlı olduğumuzun bir ölçütü olarak düşünülebilir.Cost fonksiyonu şöyle tanımlanmaktadır(Cross Entropy);

$$L(x, y) = -\sum_t y_t log(z_t)$$

Bazı Recurrent Neural Network Çeşitleri

RNN yapısının tüm çıktıları kullanılmak zorunda değildir.RNN içerisindeki gizli durum bir sonraki birime input olarak eklendiği için kullanılmayan çıkışlar tüm RNN yapısını etkilemektedir.Örnek olarak RNN'e bir veri dizisi verip çıktının en son verisi hariç kalan tüm verileri yok sayabiliriz.RNN'e bir filim hakkındaki tüm yorumları kelime kelime veri dizisi olarak verebiliriz, RNN çıkışı -1 / +1 olmak üzere beğendi / beğenmedi olabilir bu hissiyat analizi örneğidir.(sentiment analysis).Alttaki grafikte görüldüğü üzere RNN'in en son çıkışı kullanılmaktadır.

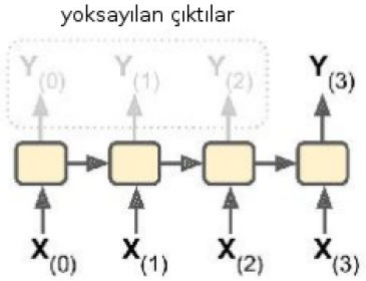

Figure 4: RNN Yapısı

RNN'in tekrar eden kısmı bir nöron yerine bir katmanda olabilir, yani bu katmanın içinde birden fazla nöron olur ve bu katman zamanda geriye doğru kopyalanır.

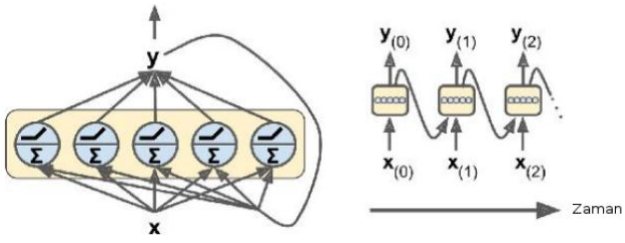

Figure 5: RNN Yapısı

Tüm RNN çeşitlerine bakacak olursak;

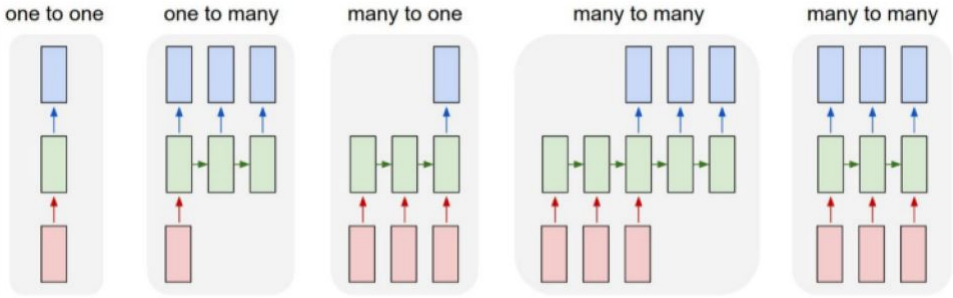

Figure 6: RNN Yapısı

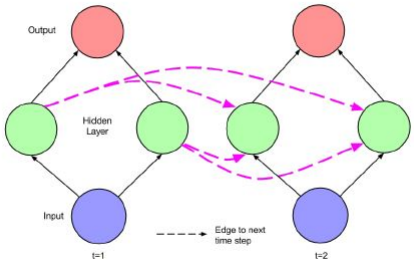

Figure 7: RNN Yapısı Zamana Göre Açılmış

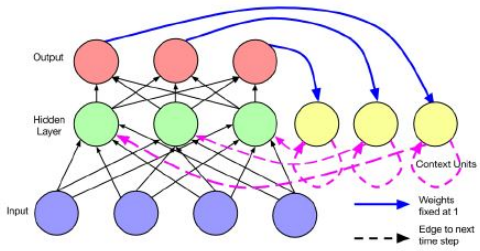

Figure 8: RNN JORDAN Yapısı

Recurrent Neural Network Eğitilmesi

RNN yapılarının eğitimi için klasik Back Propagation algoritması yerine Back Propagation Through Time (BPTT) kullanılmaktadır.Klasik Back Propagation algoritmasına göre daha zordur ve zaman bilgisini dikkate almaktadır. BPTT'ın en büyük problemi hatayı zaman adımında çok fazla geriye yayma gradyan'ın yok olmasına(vanishing) yada patlamasına(exploding) sebep olmaktadır.Tek bir giriş tek bir çıkış ve tek bir tekrarlayan gizli düğüm olduğunu düşünelim, yapı aşağıda görüldüğü gibi olacaktır.

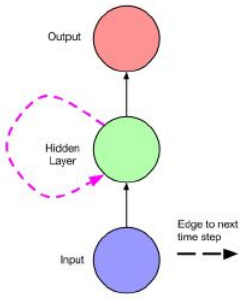

Figure 9: Basit RNN yapısı

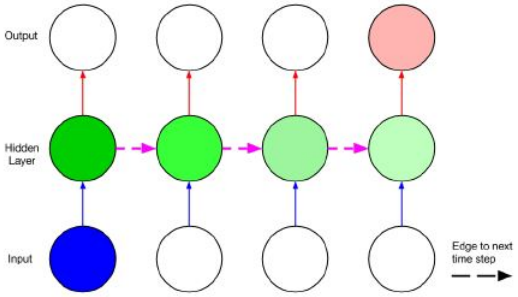

Figure 10: Basit RNN yapısının açılmış hali

Yukarıdaki grafikten de görüldüğü gibi W_{hh} tekrarlayan ağ ağırlığı için $|W_{hh}| < 1$ için gradyan bilgisi aradaki zaman aralığının bir fonksiyonu olarak hızla yok olacaktır(vanishing), öte yandan $|W_{hh}| > 1$ için hızlı bir şekilde patlayacaktır(exploding). Bu problemin önüne geçilmek için Modern RNN yapısı LSTM önerilmiştir.

Klasik NN için hata vektörü zincir kuralı ile dışarıdan içeriye doğru her bir ağ parametresinin hataya yaptığı katkıya göre güncellenmektedir.Böyle düşünüldüğü zaman klasik NN'ler iç içe geçmiş fonksiyonlar şeklindedir.
RNN'lerde iç içe olma durumu zaman faktöründen kaynaklanmaktadır.Fonksiyonlar geçmiş zaman dilimleri bağlamında iç içe geçmiş durumdadır, t anındaki çıkış geçmiş t anlarındaki fonksiyonların bir sonucu olarak görülebilir bir geri besleme durumu vardır.Her gizli durum h_t sadece bir önceki h_{t-1} 'e değil önceki tüm gizli durumlardan etkilenmektedir.Eğitim algoritması kurgulanırken bu durum dikkate alınmalıdır.

Recurrent Neural Network Kısmi Türevler
[1] $L(x, y)$ çıkış denklemini zamanda 1 adım ilerleterek $L(t+1)$ elde edelim böylece $L(t+1) = -y_{t+1}log(z_{t+1})$ elde edilir.
Geçici bir değişken tanımlayarak kısmi türevleri yazmaya başlayalım:
$\alpha_t = W_{hz}h_t + b_z$ böylece $z_t = softmax(\alpha_t)$ olacaktır.

$$\frac{\partial L}{\partial \alpha_t} = -(y_t - z_t)$$

W_{hz}: W_{hz} zamana bağlı olarak değişebilir, toplayarak ilerleyeceğiz.

$$\frac{\partial L}{\partial W_{hz}} = \sum_t \frac{\partial L}{\partial z_t}\frac{\partial z_t}{\partial W_{hz}}$$

---

b_z: Benzer şekilde yazılabilir.

$$\frac{\partial L}{\partial b_z} = \sum_t \frac{\partial L}{\partial z_t}\frac{\partial z_t}{\partial b_z}$$

W_{hh}: Türetmek icin detaylı inceleme yapalım.Zaman adımı $t \to (t+1)$ dikkate alarak hesaplamarak yapılacaktır.Bknz Figure3

$$\frac{\partial L(t+1)}{\partial W_{hh}} = \frac{\partial L(t+1)}{\partial z_{t+1}}\frac{\partial z_{t+1}}{\partial h_{t+1}}\frac{\partial h_{t+1}}{\partial W_{hh}}$$

Yukarıdaki kısmi türevler $t \to t+1$ zamanı için yazılmıştır.Fakat h_{t+1}, h_t'ye kısmen bağımlıdır. Kısmen bağımlılık için backpropagation kullanılabilir.W_{hh}, $h_t = tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$ denkleminde görüldüğü gibi tüm zaman boyunca da kullanılmaktadır(Recursive Definition).Böylece $(t-1) \to t$ için W_{hh} yazmaya devam edersek.Bknz Figure3.

$$\frac{\partial L(t+1)}{\partial W_{hh}} = \frac{\partial L(t+1)}{\partial z_{t+1}}\frac{\partial z_{t+1}}{\partial h_{t+1}}\frac{\partial h_{t+1}}{\partial h_t}\frac{\partial h_t}{\partial W_{hh}}$$

Böylece $t+1$ anından başlayarak z_{t+1}'in gradyanı $t \to 0$ anına kadar hesaplanabilir.Bu işlem Back Propagation Through Time (BPTT) olarak isimlendirilir.Bknz Figure3 Kırmızı hat boyunca.Böylece zamana bağlı olarak denklemi tekrar düzenlersek

$$\frac{\partial L(t+1)}{\partial W_{hh}} = \sum_{k=1}^{t} \frac{\partial L(t+1)}{\partial z_{t+1}}\frac{\partial z_{t+1}}{\partial h_{t+1}}\frac{\partial h_{t+1}}{\partial h_k}\frac{\partial h_k}{\partial W_{hh}}$$

Son olarak W_{hh} gradyanını zaman sırasına göre toplamını elde edelim:

$$\frac{\partial L(t+1)}{\partial W_{hh}} = \sum_t \sum_{k=1}^{t} \frac{\partial L(t+1)}{\partial z_{t+1}}\frac{\partial z_{t+1}}{\partial h_{t+1}}\frac{\partial h_{t+1}}{\partial h_k}\frac{\partial h_k}{\partial W_{hh}}$$

W_{xh}: Benzer şekilde $(t+1)$ zamanını düşünelim ve sadece $t+1$ zamanında gelen katkıyı hesaplayalım:

$$\frac{\partial L(t+1)}{\partial W_{xh}} = \frac{\partial L(t+1)}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial W_{xh}}$$

h_t ve x_{t+1} ikisi h_{t+1}'e katkı yapmaktadır.Bknz Figure3.Şekilde görüldüğü gibi h_t içinde kısmi türevlerin hesaplanması gerekmektedir.t anındaki katkıyı dikkate alırsak

$$\frac{\partial L(t+1)}{\partial W_{xh}} = \frac{\partial L(t+1)}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial W_{xh}} + \frac{\partial L(t+1)}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_t} \frac{\partial h_t}{\partial W_{xh}}$$

Böylece $t \to 0$ anına kadar olan gradyanlar toplayabiliriz.$t+1$ anı için gerekli olan gradyan hesaplanmış olur.

$$\frac{\partial L(t+1)}{\partial W_{xh}} = \sum_{k=1}^{t+1} \frac{\partial L(t+1)}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_k} \frac{\partial h_k}{\partial W_{xh}}$$

Son olarak W_{xh} gradyanını zaman sırasına göre toplamını elde edelim:

$$\frac{\partial L(t+1)}{\partial W_{xh}} = \sum_t \sum_{k=1}^{t+1} \frac{\partial L(t+1)}{\partial z_{t+1}} \frac{\partial z_{t+1}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_k} \frac{\partial h_k}{\partial W_{xh}}$$

RNN ile Lineer Olmayan Sistem Tanımlama

[2]Lineer olmayan sistem tanımlama (Nonlinear System İdentification); aktif araç süspansiyon sistemleri, hareket kontrol sistemleri, robust adaptif kontrol sistemleri vb karmaşık sistemleri veya endüstriyel süreçleri modellemek için kullanılan bir yaklaşımdır.Doğrusal olmayan sistem tanımlama, sistemin girdi ve çıktılarını kullanarak doğrusal olmayan bir sistemin matematiksel modelini tahmin etmek için kullanılan bir yöntemdir.Figür11'de sistemin blok şeması görülmektedir.

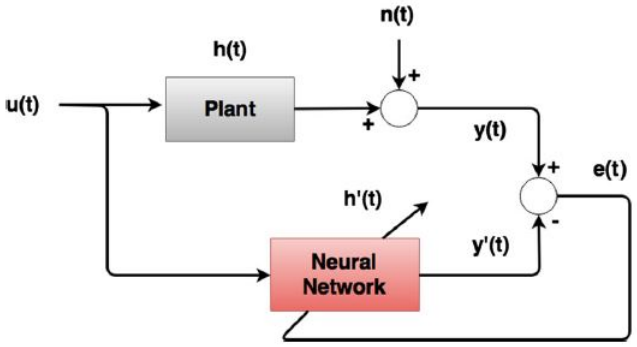

Figure 11: RNN ile Lineer Olmayan Sistem Tanımlama

Lineer olmayan sistemin fark denklemi aşağıdaki gibidir:

$$y(t) = y(t-1) - 0.25 * y(t-2) + 0.1045 * [cos(2 * u(t)) + \exp(-10 * |u(t)|)] + 0.0902 * u(t-1) + n(t)$$

Burada: $u(t)$ ve $y(t)$ sırasıyla sistemin girişi ve çıkışını ifade etmektedir. $n(t)$ sistemdeki gürültüyü ifade etmektedir.$n(t)$: 0 ortalamalı σ^2 varyanslı Gauss beyaz gürültüsünü ifade etmektedir.

Sonuçlar

RNN mimarisi 1 giriş katmanı, 1 çıkış katmanı ve 1 gizli katmandan oluşmaktadır.Klasik NN ve RNN yapıları için sonuclar aşağıdaki grafikte görülmektedir.

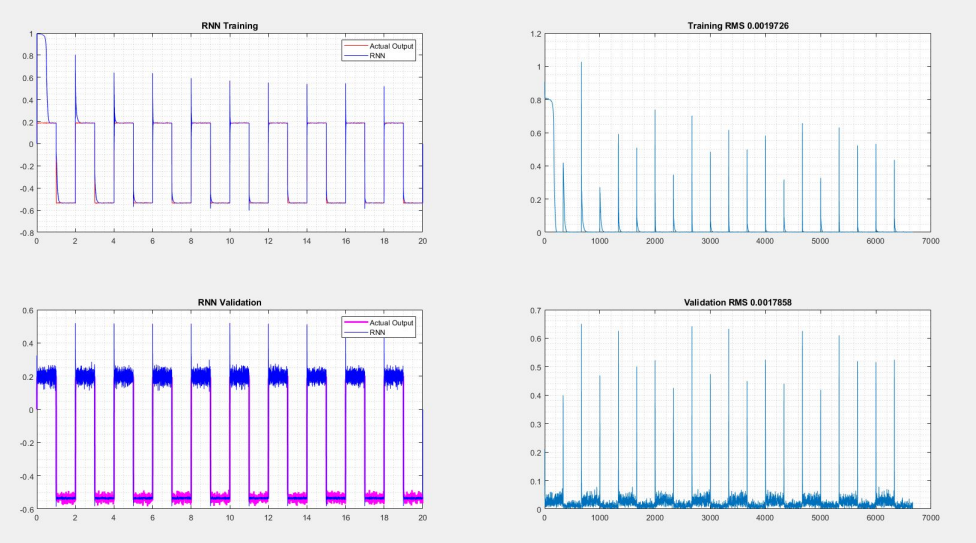

Figure 12: RNN

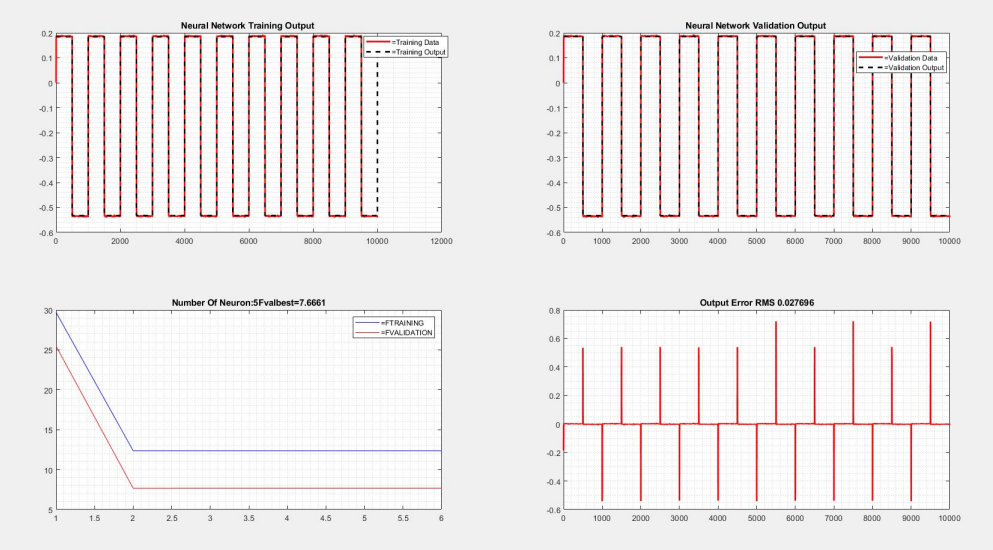

Figure 13: NN

BONUS

Fiziksel bir fenomeni modellerken yapılan yaklaşımlar genel olarak sistem tanımlama (System Identification) konusu olarak ele alınmaktadır.Sistem tanımlama, modelleme aşamasında White Box, Grey Box, Black Box olarak alt kategorilerde incelenmektedir.Oluşturulan analitik yada yapay modellerin model seçim kriteri önemli bir konudur böylece oluşturulan modele olan güveni ifade etmektedir.

Model Selection

Empirical

- Adjusted R^2 (Wherry 1931)
- Bootstrap (Efron 1979)
- Cross-validation (Stone 1974; Geisser 1975)
  - Generalized cross-validation (GCV) (Craven and Wahba 1979)
  - K-fold cross-validation
  - Leave-one-out cross-validation
- Jacknife [1]
- Linear regression
- Shibata's model selector (sms) (Shibata 1981)
- Signal-to-noise ratio
- Test set validation

Theoretical

- Akaike information criterion (AIC)
  - AIC (Akaike 1973)
  - AICc (Hurvich and Tsai 1989)
  - QAIC (Lebreton, *et al.* 1992)
  - QAICc (Lebreton, *et al.* 1992)
  - AICW (Wilks 1995)
- CAT (Parzen 1974, 1977)
- Mallows' Cp (Cp) (Mallows 1973)
- Deviance information criterion (DIC) (Spiegelhalter, *et al.* 2002)
- FIC (Wei 1992)
- Final prediction error (FPE) (Akaike 1969)
- FPEα (Bhansali and Downham 1977)
- FPEC (de Luna 1998)
- FPER (Larsen and Hansen 1994)
- GM (Geweke and Meese 1981)
- Generalized prediction error (GPE) (Moody 1991, 1992)
- Hannan and Quinn Criterion (HQ) (Hannan and Quinn 1979)
- KIC (Cavanaugh 1999)
- KICc (Cavanaugh 2004)
- Minimum description length (MDL) (Rissanen 1978)
- Minimum message length (MML) (Wallace and Boulton 1968)
- Predicted squared error (PSE) (Barron 1984)
- PRESS (Allen 1974)
- Schwarz criterion (also Schwarz information criterion (SIC) or Bayesian information criterion (BIC) or Schwarz-Bayesian information criterion) (Schwarz 1978)
- Structural risk minimization (SRM) (Vapnik and Chervonenkis 1974)
- TIC (Takeuchi's information criterion) (Takeuchi 1976)
- VC-dimension (Vapnik and Chervonekis 1968; Vapnik and Chervonenkis 1971; Vapnik 1979)

Bibliography

[1] A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation
Gang Chen.
Department of Computer Science and Engineering
University at Buffalo–SUNY

[2] A Novel Fractional Gradient-Based Learning Algorithm for Recurrent Neural Networks
Shujaat Khan, Jawwad Ahmad, Imran Naseem, Muhammad Moinuddin
© Springer Science+Business Media New York 2017

[3] Model Selection
Martin Sewell
Department of Computer Science University College London