

Yapay Zeka Dönem Projesi

YAPAY SINIR AĞLARI İLE POLİNOM DERECE TAHMİNİ

A. Bilal Güvenç 16011069

Fatih Elbasan 16011075



1. Proje Planı

a. Hedef

Değerleri verilen polinomal fonksiyonun katsayısını ve derecesini kendi geliştirdiğimiz yapay sinir ağı ile bulmak.

Katsayı bulma hedefimizi gerçekleştirememiş olmamıza rağmen derece bulmada iyi sonuçlar elde ettik.

b. Kısıtlar

Derece tahmini ve katsayı tahmini ayrı ayrı yapılacaktır, verilen “input” x_0, \dots, x_n e karşılık $y_0 \dots y_n$ olarak verilecek.

Derece tahmini kısmında “input” olarak verilen fonksiyonların derecesi maksimum olarak ağ topolojisinin “output” kısmının bir eksiği olabilir.

2. Geliştirme

a. Geliştirme Süreci

İnternet üzerinden bulduğumuz basit yapay sinir ağ gerçeklemelerine bakarak bir model geliştirdik, nöron ve ağ sınıfları oluşturup bir sistem kurduk. Geri besleme kısmında başarıyı ölçeklendiren ortalama bir başarı değeri fonksiyonu kullandık.

Sistemi polinomal fonksiyon parametreleri, çıktıları ile besleyeceğimiz için bu fonksiyonlara özel bir sınıf daha ürettik.

Yaptığımız ufak eğitimler sırasında “input” değerlerini fonksiyon sınıfından nasıl alacağımıza dair fikir elde ettik.

Denemelerimizden en yüksek başarıyı x değerlerinin 0 noktasından eşit uzaklıklarda x_0 ve x_n değerleri arasında olup eşit artış miktarlarına sahip oldukları durumlarda ulaştığımızı gördük.

Nöronlar içerisinde tanh fonksiyonu ile bir aktivasyon sağlarken “input” değerlerinin herhangi bir normalizasyon olmadan veriyorduk bu yüzden doğru tahmin oranımız çok düşüktü bunu fark edip verdiğimiz inputları $[-1, 1]$ gibi bir alana normalize ettik bu aşamadan sonra fonksiyon derecesi bulma başarı oranında gözle görülür bir yükseliş yakalandı.

Bu aşamadan sonra oluşturduğumuz rastgele fonksiyonlarla milyon iterasyonlar deneyerek sistemi eğitmeye çalıştık ancak hata miktarındaki dalgalanma ile aynı yerde takılı kalıyordu.

Bu durumu düzeltmek için sistemin topolojisini değiştirmek, input olarak verdiğimiz fonksiyon derecesini değiştirmek gibi denemeler yaptık. Denemelerimiz beklenen sonucu vermedi. Nöron sayısını arttırmak da mevcut sistemin başarısını olumsuz oranda etkiledi.

Yapay sinir ağlarında kullanılan eta ve alpha değişkenlerini ayarlayarak asıl etkiyi bu parametrelerle sağladık.

Momentumu çok yüksek bir değer öğrenme oranını ise çok düşük bir değer vererek maksimum başarıyı elde ettik.

b. Ekstra Özellikler

Milyon İterasyonlu eğitimler yaptığımızda nispeten uzun sürüyordu ve program sonlandığında mevcut modeli kaybediyorduk, bu yüzden modeli kaydetme ve geri yükleme modüllerini yazdık.

Eğitimin ileri aşamalarında eta değeri yetersiz olmaya başladı. Bunun için eğitime bağlı bir şekilde iterasyon miktarı arttıkça eta değerini düşüren bir fonksiyon ekledik.

Kaynaklar

YOUTUBE tutorials, seniors from **stackoverflow**

Ekran Görüntüleri

```
train start
generating functions
func gen finished
iter 1% avg 0.543869 eta:1.71e-05 alpha:0.995
iter 2% avg 0.246575 eta:1.539e-05 alpha:0.995
iter 3% avg 0.134361 eta:1.3851e-05 alpha:0.995
iter 4% avg 0.0729301 eta:1.24659e-05 alpha:0.995
iter 5% avg 0.0633979 eta:1.12193e-05 alpha:0.995
iter 6% avg 0.0587014 eta:1.00974e-05 alpha:0.995
iter 7% avg 0.042383 eta:9.08764e-06 alpha:0.995
iter 8% avg 0.0451868 eta:8.17888e-06 alpha:0.995
iter 9% avg 0.0346507 eta:7.36099e-06 alpha:0.995
iter 10% avg 0.0386133 eta:6.62489e-06 alpha:0.995
iter 11% avg 0.0256341 eta:5.9624e-06 alpha:0.995
iter 12% avg 0.0304658 eta:5.36616e-06 alpha:0.995
iter 13% avg 0.0629991 eta:4.82955e-06 alpha:0.995
iter 14% avg 0.0196579 eta:4.34659e-06 alpha:0.995
iter 15% avg 0.040691 eta:3.91193e-06 alpha:0.995
iter 16% avg 0.0297553 eta:3.52074e-06 alpha:0.995
iter 17% avg 0.027455 eta:3.16866e-06 alpha:0.995
iter 18% avg 0.0212742 eta:2.8518e-06 alpha:0.995
iter 19% avg 0.0181126 eta:2.56662e-06 alpha:0.995
iter 20% avg 0.0394911 eta:2.30996e-06 alpha:0.995
iter 21% avg 0.0189238 eta:2.07896e-06 alpha:0.995
iter 22% avg 0.0237228 eta:1.87106e-06 alpha:0.995
iter 23% avg 0.0235991 eta:1.68396e-06 alpha:0.995
iter 24% avg 0.0358734 eta:1.51556e-06 alpha:0.995
iter 25% avg 0.0257766 eta:1.36401e-06 alpha:0.995
iter 26% avg 0.0214054 eta:1.22761e-06 alpha:0.995
iter 27% avg 0.0361689 eta:1.10485e-06 alpha:0.995
iter 28% avg 0.0243841 eta:9.94361e-07 alpha:0.995
iter 29% avg 0.0288355 eta:8.94924e-07 alpha:0.995
iter 30% avg 0.0171978 eta:8.05432e-07 alpha:0.995
iter 31% avg 0.023473 eta:7.24889e-07 alpha:0.995
iter 32% avg 0.0235139 eta:6.524e-07 alpha:0.995
iter 33% avg 0.0252839 eta:5.8716e-07 alpha:0.995
iter 34% avg 0.0160412 eta:5.28444e-07 alpha:0.995
iter 35% avg 0.0210642 eta:4.756e-07 alpha:0.995
iter 36% avg 0.0207174 eta:4.2804e-07 alpha:0.995
iter 37% avg 0.0248969 eta:3.85236e-07 alpha:0.995
iter 38% avg 0.0291844 eta:3.46712e-07 alpha:0.995
iter 39% avg 0.0222008 eta:3.12041e-07 alpha:0.995
iter 40% avg 0.0313819 eta:2.80837e-07 alpha:0.995
iter 41% avg 0.0291217 eta:2.52753e-07 alpha:0.995
iter 42% avg 0.0201476 eta:2.27478e-07 alpha:0.995
iter 43% avg 0.0305115 eta:2.0473e-07 alpha:0.995
iter 44% avg 0.0278221 eta:1.84257e-07 alpha:0.995
iter 45% avg 0.0246177 eta:1.65831e-07 alpha:0.995
iter 46% avg 0.0401302 eta:1.49248e-07 alpha:0.995
iter 47% avg 0.0162151 eta:1.34323e-07 alpha:0.995
iter 48% avg 0.0293175 eta:1.20891e-07 alpha:0.995
iter 49% avg 0.0324479 eta:1.08802e-07 alpha:0.995
iter 50% avg 0.0359564 eta:9.79217e-08 alpha:0.995
iter 51% avg 0.0279778 eta:8.81296e-08 alpha:0.995
iter 52% avg 0.0376338 eta:7.93166e-08 alpha:0.995
iter 53% avg 0.0168901 eta:7.13849e-08 alpha:0.995
iter 54% avg 0.0212618 eta:6.42464e-08 alpha:0.995
iter 55% avg 0.0195611 eta:5.78218e-08 alpha:0.995
iter 56% avg 0.0170149 eta:5.20396e-08 alpha:0.995
iter 57% avg 0.0176387 eta:4.68357e-08 alpha:0.995
iter 58% avg 0.0161637 eta:4.21521e-08 alpha:0.995
iter 59% avg 0.0163235 eta:3.79369e-08 alpha:0.995
iter 60% avg 0.0201181 eta:3.41432e-08 alpha:0.995
```



```

iter 61% avg 0.0408782 eta:3.07289e-08 alpha:0.995
iter 62% avg 0.0215131 eta:2.7656e-08 alpha:0.995
iter 63% avg 0.0359868 eta:2.48904e-08 alpha:0.995
iter 64% avg 0.0261149 eta:2.24014e-08 alpha:0.995
iter 65% avg 0.0165195 eta:2.01612e-08 alpha:0.995
iter 66% avg 0.0299971 eta:1.81451e-08 alpha:0.995
iter 67% avg 0.0198049 eta:1.63306e-08 alpha:0.995
iter 68% avg 0.0260943 eta:1.46975e-08 alpha:0.995
iter 69% avg 0.0473851 eta:1.32278e-08 alpha:0.995
iter 70% avg 0.0279241 eta:1.1905e-08 alpha:0.995
iter 71% avg 0.0175519 eta:1.07145e-08 alpha:0.995
iter 72% avg 0.0225887 eta:9.64305e-09 alpha:0.995
iter 73% avg 0.0169752 eta:8.67874e-09 alpha:0.995
iter 74% avg 0.0310481 eta:7.81087e-09 alpha:0.995
iter 75% avg 0.0180209 eta:7.02978e-09 alpha:0.995
iter 76% avg 0.017832 eta:6.3268e-09 alpha:0.995
iter 77% avg 0.0220723 eta:5.69412e-09 alpha:0.995
iter 78% avg 0.0329879 eta:5.12471e-09 alpha:0.995
iter 79% avg 0.0285532 eta:4.61224e-09 alpha:0.995
iter 80% avg 0.0441749 eta:4.15102e-09 alpha:0.995
iter 81% avg 0.0372453 eta:3.73591e-09 alpha:0.995
iter 82% avg 0.0165792 eta:3.36232e-09 alpha:0.995
iter 83% avg 0.0201539 eta:3.02609e-09 alpha:0.995
iter 84% avg 0.0302115 eta:2.72348e-09 alpha:0.995
iter 85% avg 0.0209128 eta:2.45113e-09 alpha:0.995
iter 86% avg 0.0249354 eta:2.20602e-09 alpha:0.995
iter 87% avg 0.0164414 eta:1.98542e-09 alpha:0.995
iter 88% avg 0.0420803 eta:1.78688e-09 alpha:0.995
iter 89% avg 0.0228185 eta:1.60819e-09 alpha:0.995
iter 90% avg 0.0271826 eta:1.44737e-09 alpha:0.995
iter 91% avg 0.0360097 eta:1.30263e-09 alpha:0.995
iter 92% avg 0.0507765 eta:1.17237e-09 alpha:0.995
iter 93% avg 0.0465176 eta:1.05513e-09 alpha:0.995
iter 94% avg 0.0204866 eta:9.49619e-10 alpha:0.995
iter 95% avg 0.0299136 eta:8.54657e-10 alpha:0.995
iter 96% avg 0.0318687 eta:7.69192e-10 alpha:0.995
iter 97% avg 0.0198361 eta:6.92272e-10 alpha:0.995
iter 98% avg 0.0427899 eta:6.23045e-10 alpha:0.995
iter 99% avg 0.0259351 eta:5.60741e-10 alpha:0.995
iter 100% avg 0.0480638 eta:5.04667e-10 alpha:0.995
finished
train finished Thu May 16 22:40:01 2019
elapsed time: 116.375s
avg 0.0480638 eta:4.542e-10 alpha:0.995
3. derece
func (-20x^3)+(0x^2)+(0x^1)+(15) -> Derece:3 Tahmin:3 +
func (2x^3)+(1x^2)+(6x^1)+(4) -> Derece:3 Tahmin:3 +
func (-4x^3)+(2x^2)+(-2x^1)+(10) -> Derece:3 Tahmin:3 +
func (1x^3)+(0x^2)+(9x^1)+(-13) -> Derece:3 Tahmin:3 +
2. derece
func (4x^2)+(3x^1)+(4) -> Derece:2 Tahmin:2 +
func (1x^2)+(6x^1)+(4) -> Derece:2 Tahmin:2 +
func (-4x^2)+(2x^1)+(4) -> Derece:2 Tahmin:2 +
func (-7x^2)+(-3x^1)+(9) -> Derece:2 Tahmin:2 +
1. derece
func (20x^1)+(1) -> Derece:1 Tahmin:1 +
func (-7x^1)+(4) -> Derece:1 Tahmin:1 +
func (9x^1)+(36) -> Derece:1 Tahmin:1 +
func (-7x^1)+(-3) -> Derece:1 Tahmin:1 +
0. derece
func (30) -> Derece:0 Tahmin:0 +
func (-43) -> Derece:0 Tahmin:0 +
func (3) -> Derece:0 Tahmin:0 +

```

```

iter 99% avg 0.0129152 eta:5.60741e-10 alpha:0.997
iter 100% avg 0.021498 eta:5.04667e-10 alpha:0.997
finished
train finished Thu May 16 22:54:40 2019
elapsed time: 118.413s
avg 0.021498 eta:4.542e-10 alpha:0.997
Topology ->
20
30
4
-----
3. derece
func (-20x^3)+(0x^2)+(0x^1)+(15)      -> Derece:3 Tahmin:3 +
func (2x^3)+(1x^2)+(6x^1)+(4)        -> Derece:3 Tahmin:3 +
func (-4x^3)+(2x^2)+(-2x^1)+(10)     -> Derece:3 Tahmin:3 +
func (1x^3)+(0x^2)+(9x^1)+(-13)      -> Derece:3 Tahmin:3 +
2. derece
func (4x^2)+(3x^1)+(4)               -> Derece:2 Tahmin:2 +
func (1x^2)+(6x^1)+(4)               -> Derece:2 Tahmin:2 +
func (-4x^2)+(2x^1)+(4)              -> Derece:2 Tahmin:2 +
func (-7x^2)+(-3x^1)+(9)             -> Derece:2 Tahmin:2 +
1. derece
func (20x^1)+(1)                     -> Derece:1 Tahmin:1 +
func (-7x^1)+(4)                     -> Derece:1 Tahmin:1 +
func (9x^1)+(36)                     -> Derece:1 Tahmin:1 +
func (-7x^1)+(-3)                    -> Derece:1 Tahmin:1 +
0. derece
func (30)                            -> Derece:0 Tahmin:0 +
func (-43)                           -> Derece:0 Tahmin:0 +
func (3)                             -> Derece:0 Tahmin:0 +
Press any key to continue . . .

```