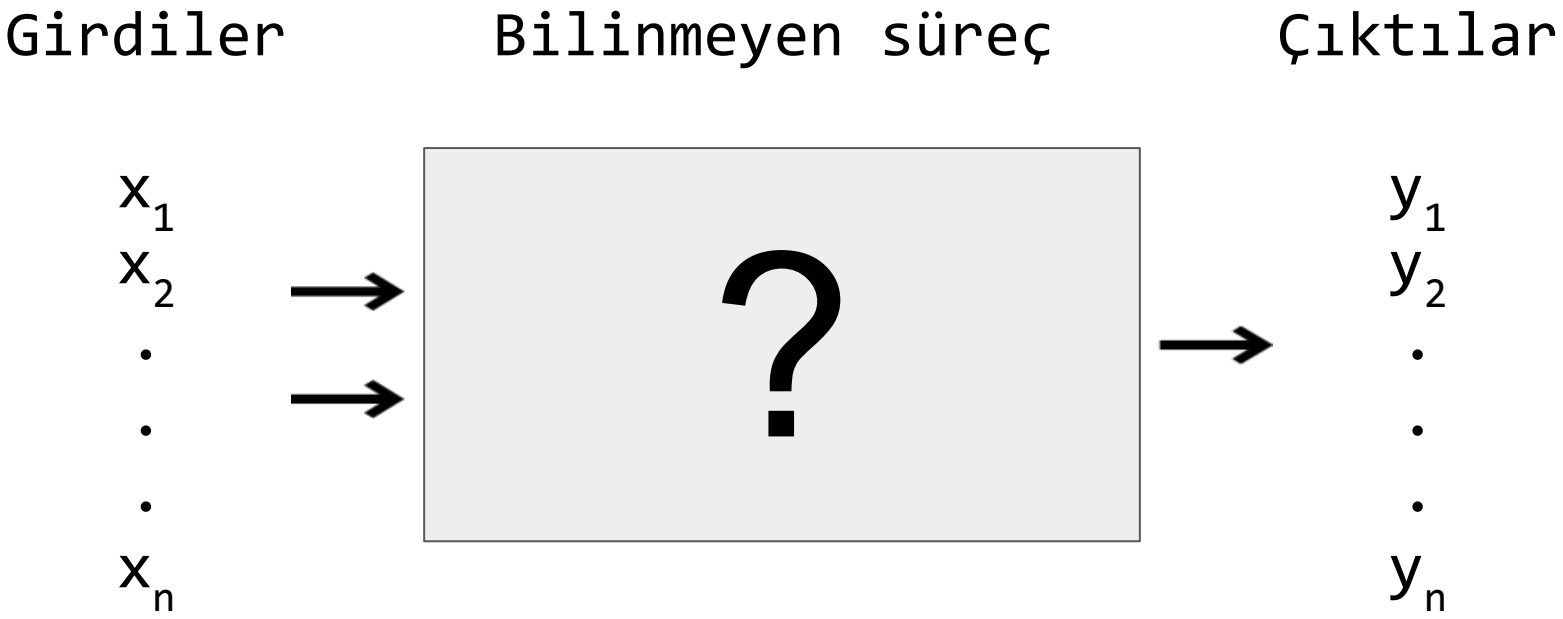
Derin Öğrenmeye Giriş

Deniz Yuret
2018

5 slaytta yapay öğrenme

Yapay öğrenme: gözlemleme

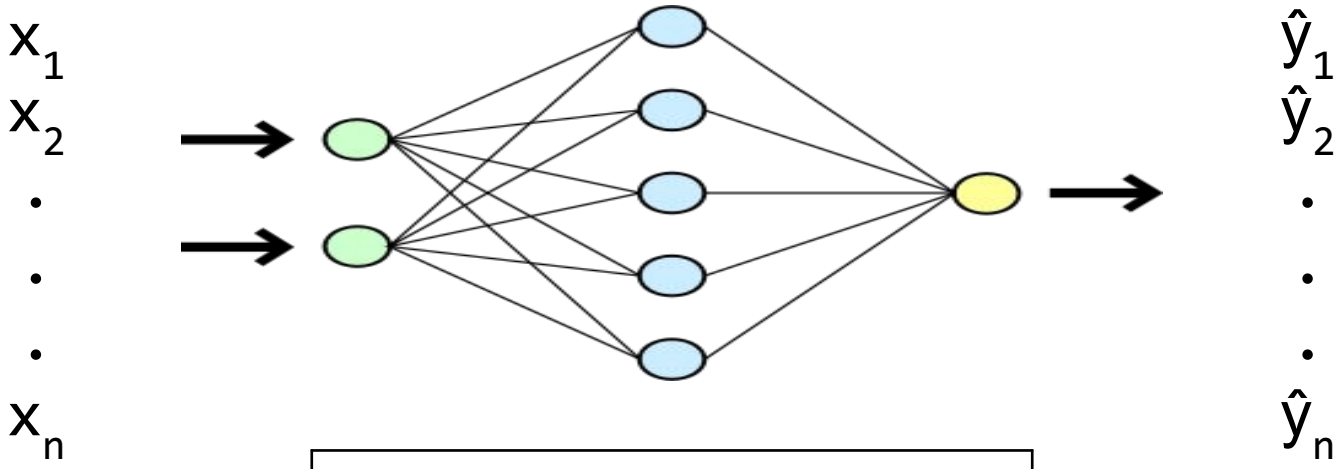


Yapay öğrenme: modelleme

Girdiler

Türevlenebilir program

Tahminler

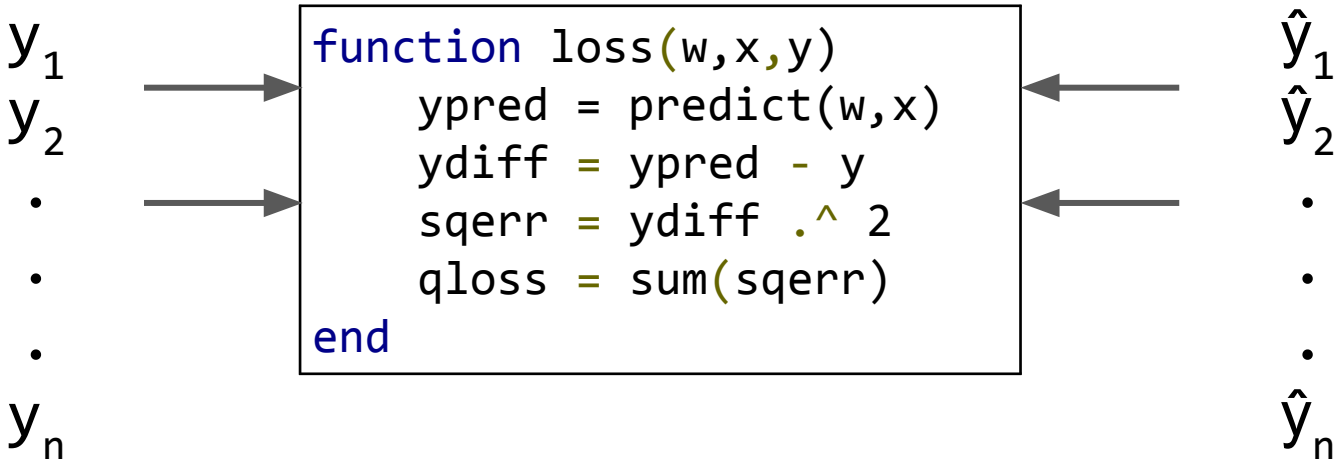



```
ypred = predict(w, x)
```

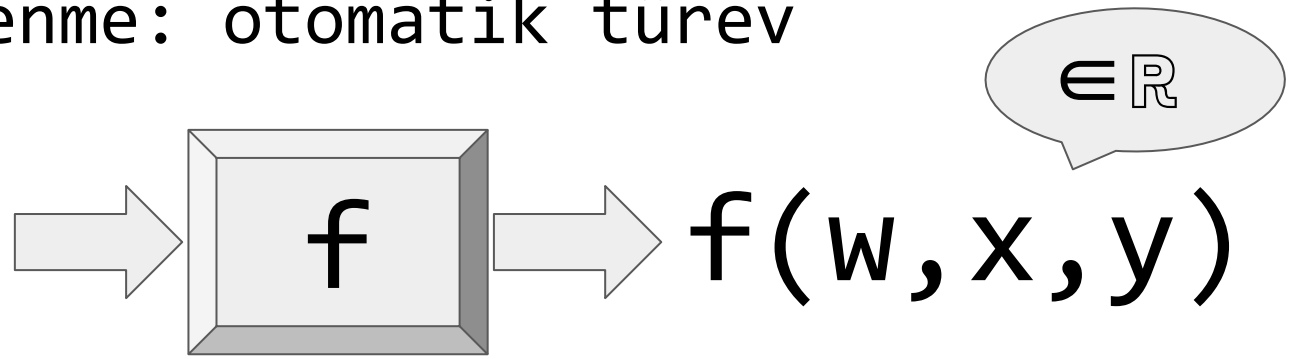
Yapay öğrenme: hata (loss) fonksiyonu

Çıktılar

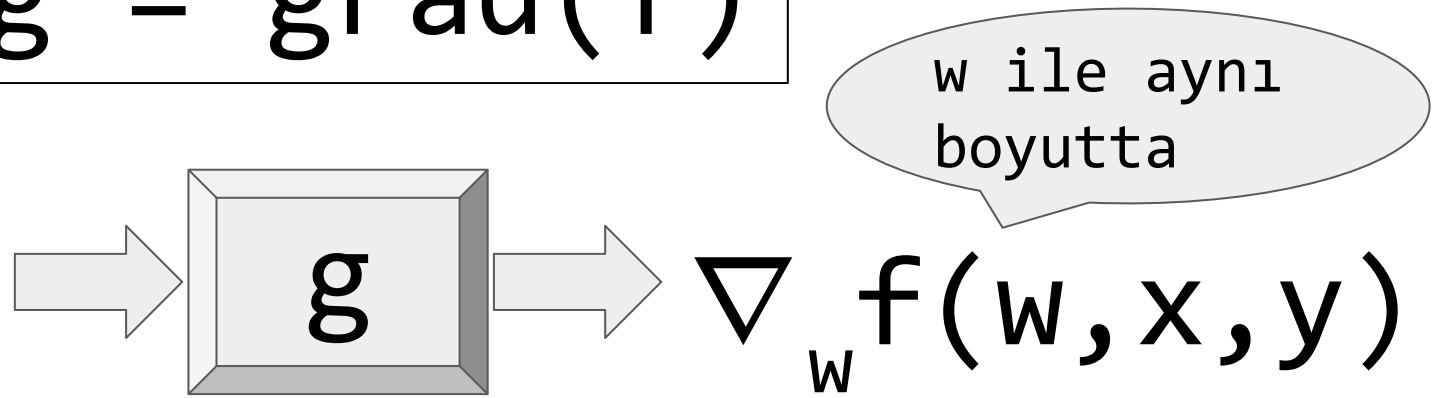
Tahminler



Yapay öğrenme: otomatik türev



$$g = \mathrm{grad}(f)$$



Yapay öğrenme: optimizasyon döngüsü

- w parametreleri başlangıçta rastgele seçilir
- loss(w,x,y) => w ile yapılan tahmindeki hata
- gfun(w,x,y) => loss'un w parametrelerine göre türevi
- data = [(x1,y1),(x2,y2),...]: eğitim verisi
- SGD(w,data,loss) => hatayı azaltan w parametreleri bulur

```
function SGD(w, data, loss)
    gfun = grad(loss)
    for (x,y) in data
        g = gfun(w, x, y)
        w = w - g * learningRate
    end
    return w
end
```

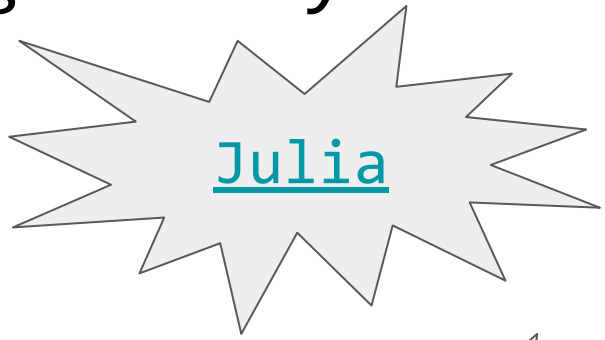
Derin öğrenme nedir?

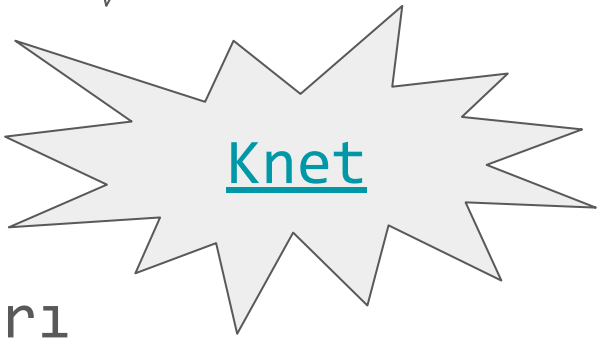
Çok katmanlı modellerle yapılan yapay öğrenme

(Bkz. http://josephpcohen.com/w/visualizing-cnn-architectures-side-by-side-with-mxnet)

Derin öğrenme için gerekli yazılım

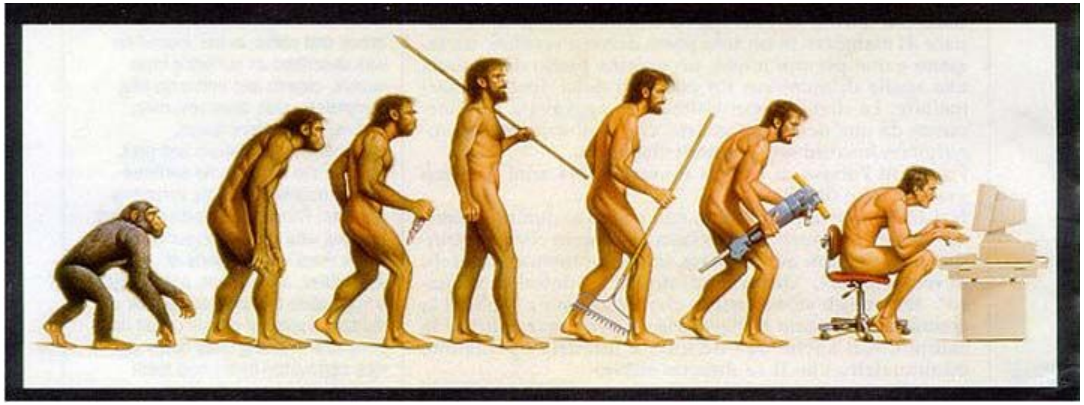
- Modelleme dili
- GPU desteği
- Otomatik türev
- Optimizasyon algoritmaları





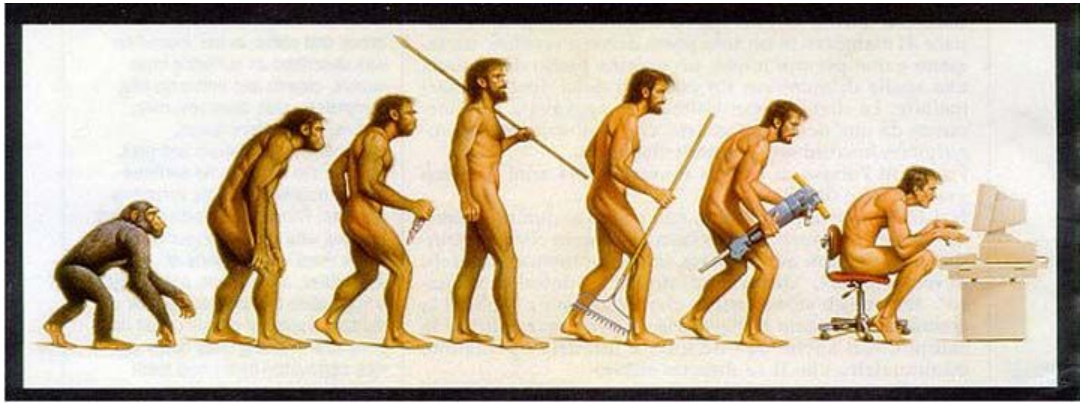
Neden Julia ve Knet?

Bilgisayar dillerinin evrimi



Machine Code → Assembler → Fortran BASIC → C ⋯⋯► Julia

Derin öğrenme
~~Bilgisayar~~ dillerinin evrimi



Machine Code

Assembler

Fortran
BASIC

C

Julia

CUDA
ConvNet

Caffe

Torch
Theano
TFlow

PyTorch
DyNet
TF Fold

Knet.jl nedir?

Julia + grad + küçük ekler!

Modelleri sınırlı bir mini-dil yerine Julia'da ifade edelim

küçük ekler =

gpu support, custom memory management, convolution library, efficient gpu array kernels for broadcasting, reduction, indexing, concatenation and their gradients...

5 dakikada Julia

Doğrusal cebir

```
julia> a
3x3 Array{Int64,2}:
 10  40  70
 20  50  80
 30  60  90

julia> b
3-elt Array{Int64,1}:
 1
 2
 3
```

```
julia> a * b
3-elt Array{Int64,1}:
 300
 360
 420

julia> b' * a
1x3 Array{Int64,2}:
 140  320  500
```

Tekli dizi işlemleri

```
julia> c
3x3 Array{Int64,2}:
 1  4  7
 2  5  8
 3  6  9
```

```
julia> log.(c)
3x3 Array{Float64,2}:
 0.0       1.38629  1.94591
 0.693147  1.60944  2.07944
 1.09861   1.79176  2.19722

julia> exp.(c)
3x3 Array{Float64,2}:
  2.71828   54.5982  1096.63
  7.38906  148.413   2980.96
 20.0855   403.429   8103.08
```

İkili dizi işlemleri

```
julia> a
3x3 Array{Int64,2}:
 10  40  70
 20  50  80
 30  60  90

julia> c
3x3 Array{Int64,2}:
 1  4  7
 2  5  8
 3  6  9
```

```
julia> a + c
3x3 Array{Int64,2}:
 11  44  77
 22  55  88
 33  66  99

julia> a * c
3x3 Array{Int64,2}:
 300  660  1020
 360  810  1260
 420  960  1500
```

Yayılan (broadcasting) dizi işlemleri

```
julia> a
3x3 Array{Int64,2}:
 10  40  70
 20  50  80
 30  60  90

julia> b
3-elt Array{Int64,1}:
 1
 2
 3
```

```
julia> a + b
ERROR: DimensionMismatch

julia> a .+ b
3x3 Array{Int64,2}:
 11  41  71
 22  52  82
 33  63  93
```

Dizi indirgeme

```
julia> c
3x3 Array{Int64,2}:
 1  4  7
 2  5  8
 3  6  9
```

```
julia> sum(c)
45

julia> sum(c,1)
1x3 Array{Int64,2}:
 6  15  24

julia> sum(c,2)
3x1 Array{Int64,2}:
 12
 15
 18
```

Dizi indisleme

```
julia> a
3x3 Array{Int64,2}:
 10  40  70
 20  50  80
 30  60  90
```

```
julia> a[1,2]
40

julia> a[5]
50

julia> a[1:2,2:3]
2x2 Array{Int64,2}:
 40  70
 50  80

julia> a[1,:]
1x3 Array{Int64,2}:
 10  40  70
```

Dizi bitiştirme

```
julia> a
3x3 Array{Int64,2}:
 10  40  70
 20  50  80
 30  60  90

julia> b
3-elt Array{Int64,1}:
 1
 2
 3
```

```
julia> [a b]
3x4 Array{Int64,2}:
 10  40  70  1
 20  50  80  2
 30  60  90  3

julia> [a;b']
4x3 Array{Int64,2}:
 10  40  70
 20  50  80
 30  60  90
  1   2   3
```

Fonksiyonlar

```
function loss(w, x, ygold)
    ypred = w * x
    ydiff = ypred - ygold
    sqerr = ydiff .^ 2
    qloss = sum(sqerr)
    return qloss
end

# veya kısa tanım ile:
loss(w,x,y)=sum((w*x-y).^2)
```

Demo

Derin öğrenme IJulia defterleri

- Julia ne kadar hızlı?
- Julia öğrenelim
- MNIST el yazısı tanıma problemi
- Doğrusal modeller, türev ve optimizasyon
- Çok katmanlı modeller
- Konvolüsyonel modeller
- Özyinelemeli modeller
  - sentiment analizi, dil modelleri, tercüme