Lecture 7: Neural Nets and the Learning Function

Agenda

- Construction of Neural Nets
- Distance Matrices

Source: Sections VII.1 and IV.10 in Linear Algebra and Learning from Data (2019) by **Gilbert Strang**

Construction of Neural Nets

- Given 0, 1,…9 -> 10 digits, one of them is drawn in a square
- How computer recognizes which this digit is? This is ML problem. Learn from examples is the idea to start with
- Take M different images, each image having p-pixels and it is represented $v=(v_1, \ldots v_p)$ and it will say about image how it is, say, dark or light?
- Each M^{th} image having p-features, M vectors v in p-dimensional space
- For every v in training set we have those digits it represent
- M inputs in R^p each with an output from 0 to 9
- Need a rule: ML proposes a rule that succeeds on (most of) the training images
- ”Succeed" means : this rule gives the correct digit for a much wider set of test images, taken from the same population
- This essential requirement is called *generalization*
- First answer : *F(v)* could be a linear function from R^p to $\mathbf{R}^{10}$ (10 by *p* matrix)
- 10 outputs the probabilities of the numbers 0 to 9
- 10p entries and *M* training samples
- Linear function is too limited, and the input-output rule is nowhere near linear

Source: Section VII.1 in Linear Algebra and Learning from Data (2019) by Gilbert Strang

Construction of Neural Nets

Deep learning is Continuous Piecewise Linear (CPL) functions.
Linear for simplicity, continuous to model an unknown but reasonable rule, and
piecewise to achieve the nonlinearity that is an absolute requirement for real images and data

Here is a first construction of a piecewise linear function of the data vector v.
Choose a matrix A_1 and vector b_1.
Set to zero (this is the nonlinear step) all negative components of $A_1 v + b_1$.
Multiply by a matrix A_2 to produce 10 outputs in $w = F(v) = A_2(A_1 v + b_1)_+ \ldots + A_{10}(A_{10} v + b_{10})$
Vector $(A_1 v + b_1)_+$.. forms a "hidden layer" between the input v and the output w.

Source: Section VII.1 in Linear Algebra and Learning from Data (2019) by Gilbert Strang

Construction of Neural Nets

Learning Function F (x,v) where x are the weights, and v are the feature vectors, the sample feature vectors (training dataset)

So those feature vectors v_0 come from the training data, either one at a time, if we're using stochastic gradient descent (discussed detail later) with mini-batch size 1

Or B at a time, if we're doing mini-batch of size B, or the whole thing, a whole epoch at once, if we're doing full-scale gradient vector

So those are the feature vectors, and these are the numbers in the linear steps, the weights

Weight matrix A_K multiply by v and bias vectors b_K that adds on to shift the origin. **Optimize x and v**

Take first step of learning Function F:

v_1=ReLU $(F(A_1,b_1, v_0))$ → non-Linear Step

ReLU (Rectified Linear Unit) is an activation function.

Construction of Neural Nets

$(A_1 v_0 + b_1) \rightarrow$ linear step
$v_1 = \mathrm{ReLU}\,(A_1 v_0 + b_1)$ non-linear step

Generally, $v_k = \mathrm{ReLU}\,(A_k v_{k-1} + b_k)$ where $k=1, \ldots l$ (l is number of layers)

1-Training sample

$$v_1 = \mathrm{ReLU}\left(A_1 \begin{bmatrix} v_0 \\ \\ \\ \end{bmatrix} + b_1 \right)$$

$v_l = A_l v_{l-1} + b_l$

Don't do ReLU at the last layer, so it's just $A_l\, v_{l-1} + b_l$.

So this picture is clearer to distinguish between the weights

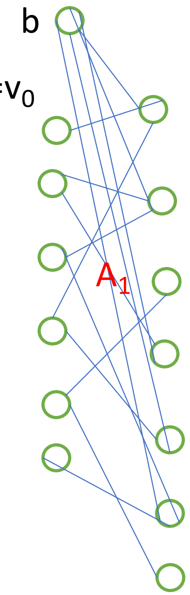
$x = A_1, b_1, A_2, b_2 \ldots . A_l . b_l$

x stands for all the weights that we compute up to A_l, b_l, so that's a collection of all the weights

Often weights x's are undetermined because the number of x's in A's, b's are greater than the number of v's in the training sets

Construction of Neural Nets

1^{st} layer: $v_1 = A_1 v_0 + b_1$



Choose x to min Loss Function L= $(\frac{1}{N})\ [\sum_{i=1}^{N} F(x, vi)$

$L(x) = (\frac{1}{N})\ [\sum_{i=1}^{N} F(x, vi) - \text{truei}]$

Question: do we use the whole function L at each iteration, or do we just pick only b, of the samples to look at iteration number k?

This is the L(x) then added up over all v's and the neural net produces. It's supposed to be close to the 'true'

Popular Loss functions:

(1) Square loss= sum of $\| \|_2^2$ → regression

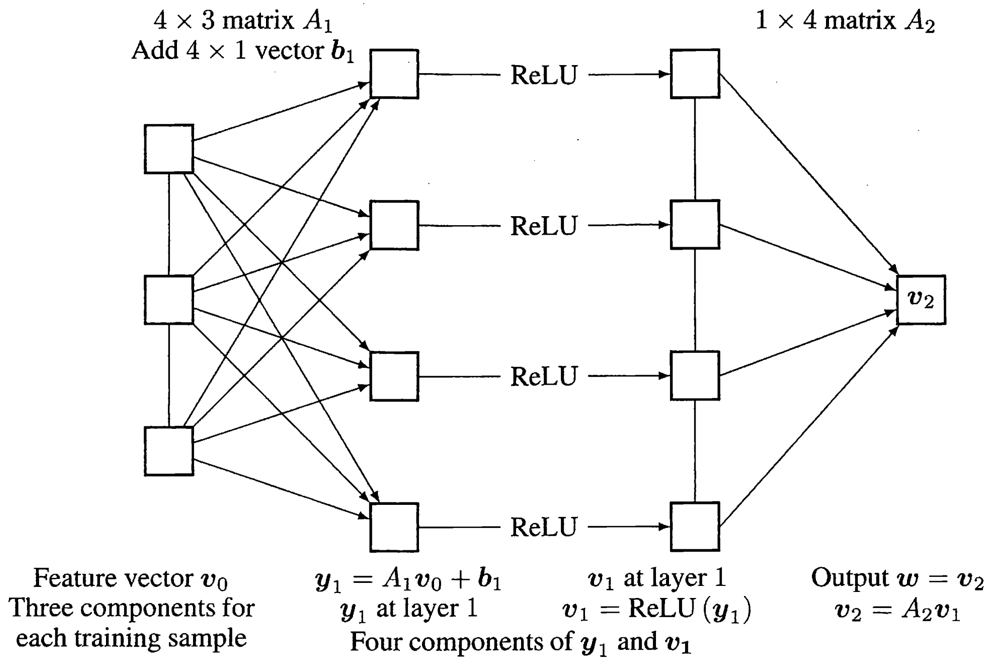
(2) L^1 loss = sum of $\| \|^1$ → Lasso

(3) Hinge loss (-1,1 Classification)

(4) Cross-entropy loss (neural nets)

Construction of Neural Nets

The goal in optimizing $\boldsymbol{x} = A_1, \boldsymbol{b}_1, A_2, \boldsymbol{b}_2$ is that the output values $\boldsymbol{v}_\ell = \boldsymbol{v}_2$ at the last layer $\ell = 2$ should correctly capture the important features of the training data $\boldsymbol{v}_0$.



The output v_2 (plus or minus) classifies the input v_o (object 1 (say, brain tumor) or Object 2(say, no brain tumor). Then v_2 is a composite measure of the 3-component feature vector v_0 . This net has 20 weights in A_k and b_k.

Source: Section VII.1 in Linear Algebra and Learning from Data (2019) by Gilbert Strang

Distance Matrices

Question: distances squared:: $\|x_i - x_j\|^2 = d_{ij}$. Find positions x_j in R^d (also find d i.e, the dimension of the space).

$\|x_i - x_j\|^2 =$ given d_{ij} . Find x's. Given D ={d_{ij}} distances matrix, to find X matrix which gives the positions

In machine learning, you're given a whole lot of points in space, feature vectors (points) in a high-dimensional space, and those are related i.e., they are connected.

They tend to fit on a surface in high-dimensional space, a low- dimensional surface in high-dimensional space

Let's recognize the connection between distances and positions:

$$D = d_{ij} = \|x_i - x_j\|^2 = (x_i - x_i)^T (x_i - x_j) = x_i^T x_i - x_i^T x_j - x_j^T x_i + x_j^T x_j \text{ (entries in D)} \qquad (1)$$

$x_i^T x_i$ produces a matrix with constant rows (no dependence on j). $x_j^T x_j$ produces a matrix with constant columns (no dependence on i) (For detail See Appendix foil)

$\|x_i\|^2$ and $\|x_j\|^2$ in both of those matrices are on the main diagonal of $G = X^T X$

Those are the numbers in the column vector diag(G) (For detail See Appendix foil 12)

Distance Matrices

Middle terms $-2x_i^T x_j$ in (1) in last foil, $-2G = -2X^T X$

*R*ewrite (1) as an equation for the matrix *D,* using the symbol **1** for the column vector of *n* ones

That gives constant columns and $\mathbf{1}^T$ gives constant rows

So, $D = \mathbf{1}\,\mathrm{diag}(G)^T - 2G + \mathrm{diag}(G)\,\mathbf{1}^T$ (2)

(Note: deduction of equation 2 is given in the next foil and For detail See Appendix in last foil)

Given D Find X // actually find $X^T X = G$ then find X from G

We'll find $X^T X$

Because we have dot products of X's. Find out what $x_i . x_j$ is.

Let's call this matrix G for the dot product matrix, and then find X from G.

Now let us say diagonal matrix $D_{ii} = (x_i, x_i)$. Let us write an equation for G with dot matrix $X^T X$

Distance Matrices

Solve equation (2) (from last foil) for $G = X^T X$

Place first point at the origin : $x_1 = 0$. For every $\|x_i - x_1\|^2$ is $\|x_i\|^2$
First column d_i of D (which is given) is the same as

$\mathrm{diag}(X^TX) = \mathrm{diag}(G) = (\|x_1\|^2, \|x_2\|^2, \dots, \|x_n\|^2) \rightarrow \mathrm{diag}(G) = d_1$ and $\mathrm{diag}(G)\,\mathbf{1}^T = d_1\mathbf{1}^T$

$$X^TX = G = -\tfrac{1}{2} D + \tfrac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \underbrace{\begin{pmatrix} d \end{pmatrix}^T}_{\text{Every colm}} + \tfrac{1}{2} \underbrace{\begin{pmatrix} d \end{pmatrix}}_{\text{Every row}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}^T$$

Now G comes from D. G is positive semidefinite provided the distances in D satisfy the triangle inequality

Ref:Menger: *Amer. J. Math.* 53; Schoenberg: *Annals Math.* 36

This is the key equation

Matrix form

$$D = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}^T + \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}^T - 2XX^T$$

$$XX^T = \tfrac{1}{2}\left[D - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} d_1 \\ : \\ d_i \end{pmatrix}^T - \begin{pmatrix} d_1 \\ : \\ d_i \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}^T\right]$$

Distance Matrices

Given XX^T Find X (nxn)

Find X up to an orthogonal transformation, as XX^T is symmetric

Two leading candidates

XX^T is positive or semidefinite, this is semidefinite. Given a semidefinite matrix and find a square root. Matrix is the XX^T and find X

(1) Evaluate of $XX^T = Q\Lambda Q^T$
(2) Elimination on XX^T

There are many candidates, because if you find one i.e., any QX .
Because Q^TQ in there, it's the identity.

Distance Matrices

(1) Take $X=Q\sqrt{\Lambda}Q^T=X^T$
$XX^T= (Q\sqrt{\Lambda}Q^T)(Q^T\sqrt{\Lambda}Q)= \Lambda=$ I=identity matrix

(2) Elimination on XX^T = LDU (L, a lower triangular, times D, the pivots, times U, the upper triangle)
= LDL^T (U is replaced by L^T)

Then $X= \sqrt{D}L^T$ (This is the Cholesky Factorization)

(Note: when X^TX, then X^TX coming correctly. X^T will be L^T. Transpose will give L. Square root of D will be $\sqrt{D}$. We'll give the D, and then the L^T is right)

Appendix

$D = d_{12} = ||x_1 - x_2||^2 = (x_1 - x_2)^2 = x_1^2 - 2\, x_1 x_2 - x_2^{\,2} = x_1^T.x_1 - 2\, x_1^T x_2 + x_2^T.x_2$

$x_1^T.x_1 = \begin{pmatrix} x_1 \\ 0 \end{pmatrix} \begin{pmatrix} x_1 & 0 \end{pmatrix}$ and $2x_1^T.x_2 = 2 \begin{pmatrix} x_1 \\ 0 \end{pmatrix} \begin{pmatrix} x_2 & 0 \end{pmatrix}$ and $x_2^T.x_2 = \begin{pmatrix} x_2 \\ 0 \end{pmatrix} \begin{pmatrix} x_2 & 0 \end{pmatrix}$

$\begin{pmatrix} x_1 \\ 0 \end{pmatrix} \begin{pmatrix} x_1 & 0 \end{pmatrix} = \mathbf{1}\ \mathbf{diag} \left[\begin{pmatrix} x_1 \\ 0 \end{pmatrix} \begin{pmatrix} x_1 & 0 \end{pmatrix} \right]^{\mathbf{T}} = \mathbf{1} \begin{pmatrix} x_1.x_1^T \\ \mathbf{0} \end{pmatrix}^{\mathbf{T}}$

Similarly,

$D = d_{12} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \mathbf{diag} \begin{pmatrix} x_1^2 \\ \mathbf{0} \end{pmatrix}^{\mathbf{T}} \mathbf{-}\ 2 \begin{pmatrix} x_1 \\ 0 \end{pmatrix} \begin{pmatrix} x_2 & 0 \end{pmatrix} \mathbf{+\ diag} \begin{pmatrix} x_1^2 \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\mathbf{T}}$

$D = \mathbf{1}\ \mathrm{diag}(G)^T - 2G + \mathrm{diag}(G)\ \mathbf{1}^T$ where $G = x_1^T.x_2$