

انهارده هنتكلم علي النويرال نتوركس .. بس بشكل جد بقا .. احنا قلنا المره اللي فاتت اننا نقدر نبني ال recipe for machine learning .. في الحاله ديه كان المقصود هو الصوره ديه



Background

A Recipe for Machine Learning

1. Given training data:

$$\{x_i, y_i\}_{i=1}^N$$

Face
Face
Not a face

2. Choose each of these:

- Decision function

$$\hat{y} = f_{\theta}(x_i)$$

- Loss function

$$\ell(\hat{y}, y_i) \in \mathbb{R}$$

Examples: Linear regression, Logistic regression, Neural Network

Examples: Mean-squared error, Cross Entropy

5

بيكون عندك تريننج داتا و انت بتختار ال decision function وال loss function وبعدين تقفل ال empirical risk عن طريق انك تعمل تريننج بال SGD .. ده اللي احنا عملناه في كل المحاضرات .. انهارده هنتكلم علي تعريف مختلف لل $f_{\theta}(x_i)$...

Goals for Today's Lecture

1. Explore a **new class of decision functions** (Neural Networks)

2. Consider **variants of this recipe** for training

2. Choose each of these:

- Decision function

$$\hat{y} = f_{\theta}(x_i)$$

- Loss function

$$\ell(\hat{y}, y_i) \in \mathbb{R}$$

Train with SGD:

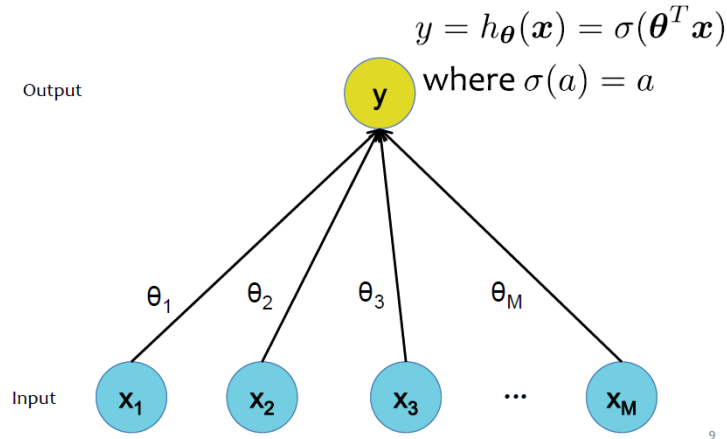
Take small steps opposite the gradient)

$$\theta^{(t+1)} = \theta^{(t)} - \eta_t \nabla \ell(f_{\theta}(x_i), y_i)$$

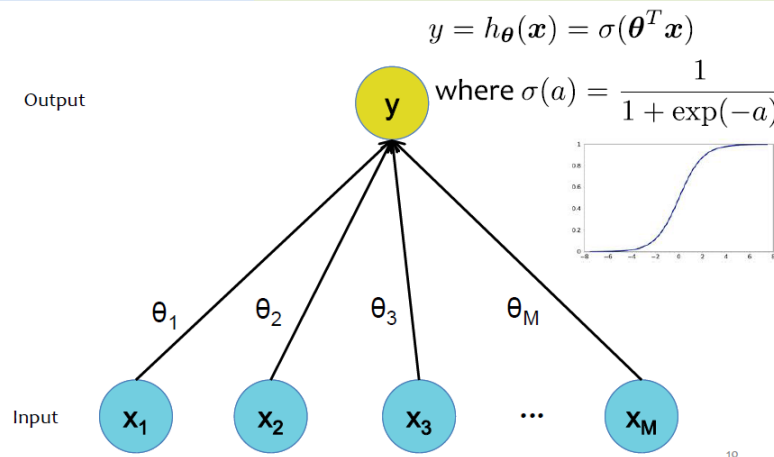
احنا قلنا ان ال linear regression عندنا فيتشرز هي ال inouts و عندنا ال weights علي ال edges زي اللي في الصوره تحت كذا و في صوره بتوضح ال dot product ما بين ال فيتشرز وما بين ال weights و غالبا بيبقا عندك ال non-linear function الي هي السجمويد .. بيعدي عليها ال dot product ..

احنا نقدر نغير ال sigmoid ونحط مكانها sign function وده هيقا ال perceptron classifier ومشكلته انو مش بي fit في الوصفه بتاعت الماشين ليرننج عشان ال sign function مش differentiable .. ال analogy ما بين ال sigmoid و ال sign function هو اللي جاب الترم بتاع ال multi-layer perceptron .. اسم تاني لأنواع النويرال نتوركس اللي هنتكلم عنها انهارده ..

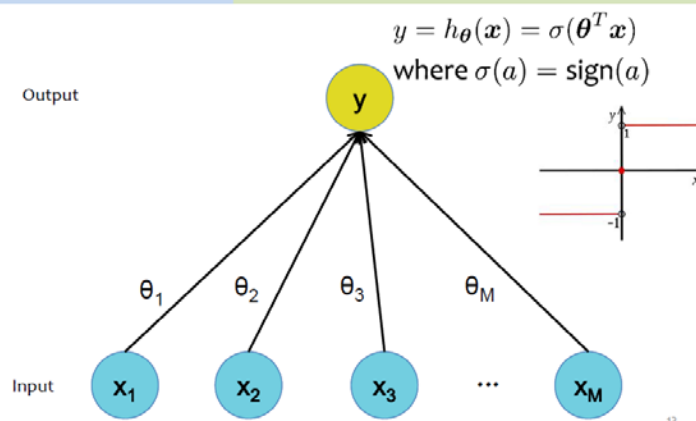
Decision Functions Linear Regression



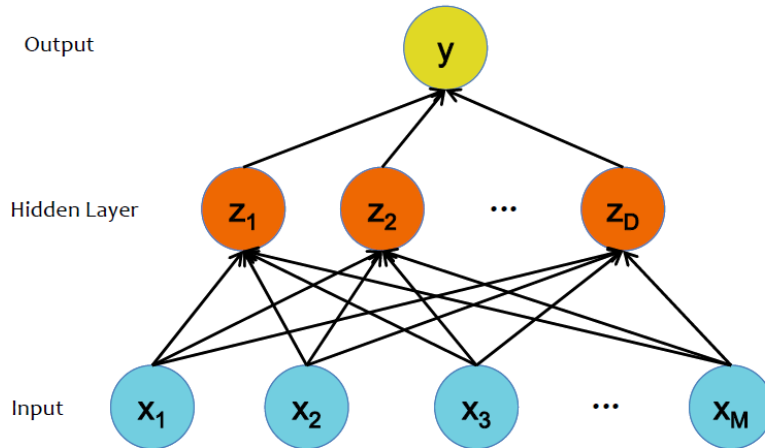
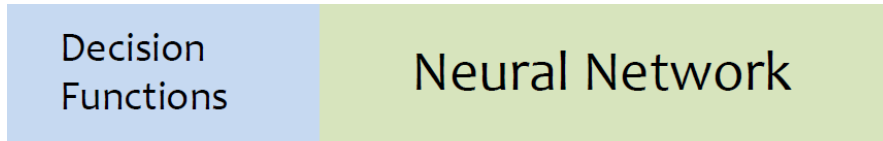
Decision Functions Logistic Regression



Decision Functions Perceptron

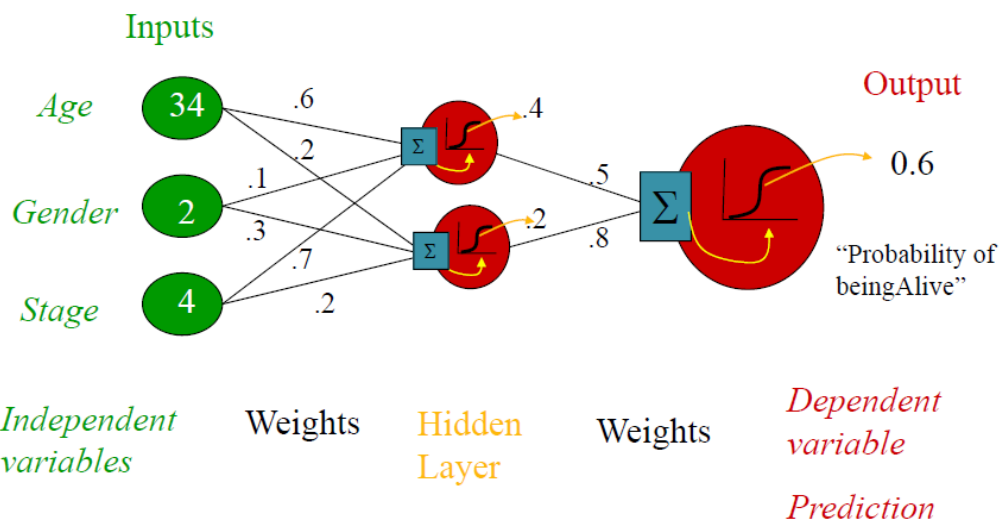


المهم .. الصورة الجايه هي النويرال نتورك بتاعتنا والفكره بتاعت النويرال نتورك هو اننا نخط شويه logistic regression models كثيره مع بعض ..
عشان بيقا عندنا فانكشن جديده .. decision function جديده يعني ..

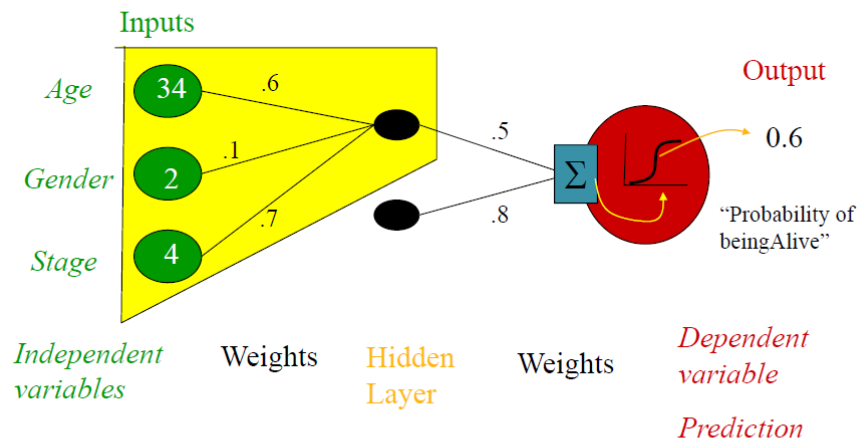


عندك هنا z_1 وهي أول logistic regression .. وفي اسهم كثيره داخله علي ال z_1 .. فهي فانكشن في ال x زي ال logistic regression .. و z_2 برضو ليها اسهم داخله عليها .. بس الويٲٲس بتاعتهم مختلفه عن الي داخلين علي z_1 .. انت عاوز تحسب ال y .. خذ ال z_s اللي عندك و اعمل دوت بروضت مع ال weights المقابل ليهٲ .. كذا اللي عملناه دلوقت اننا عرفنا فانكشن معقده أكثر من ال dot product and sigmoid ... تعال نبص علي تفاصيل اكثر شويه ..

Neural Network Model

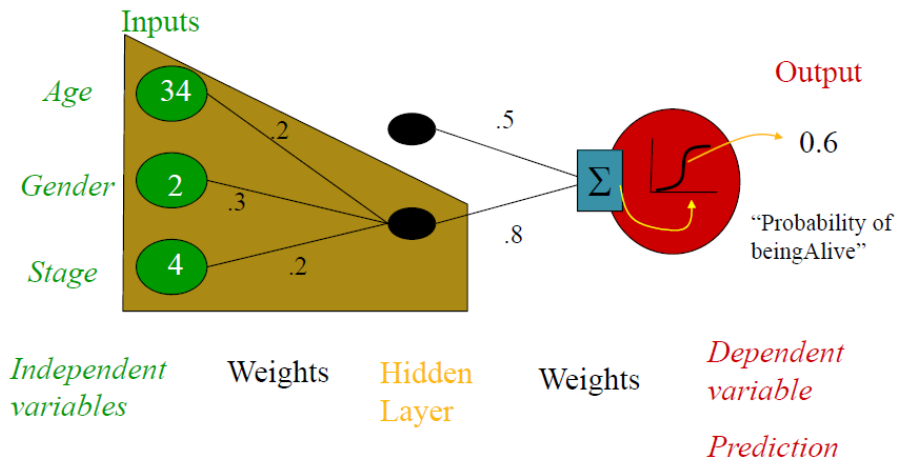


“Combined logistic models”



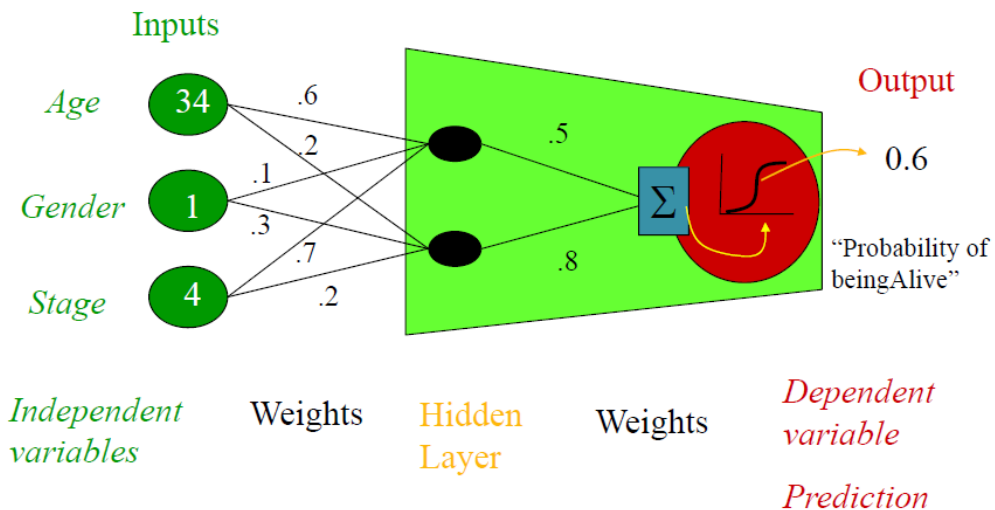
© Eric Xing @ CMU, 2006-2011

16



© Eric Xing @ CMU, 2006-2011

17

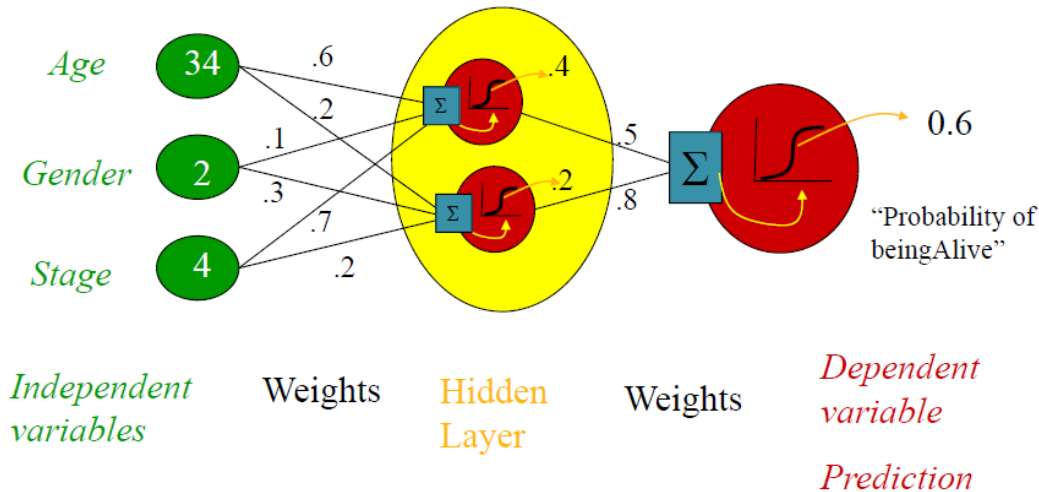


© Eric Xing @ CMU, 2006-2011

18

خلينا نسأل السؤال .. هل ده فعلاً زي شوية logistic regression ... "مسا مسا يا شوية كابستورز و المجلس الأعلى للكاپستورز" .. الإجابة .. لا الحقيقة مش بالطبط .. عشان في اللوجستيك ريجريشن انت بتفترض دايماً أنك عارف ايه هي ال correct value of the target ... بالنسبة لل training data بتاعتنا .. احنا عارفين ايه هو ال input بتاعنا وعارفين ايه هو ال output اللي في الآخر .. انما مش هنبقا عارفين ايه هي الارقام بتاعت ال z اللي في الصور اللي فوق .. ال z هنبقا مستخدمه عشان تبني complicated function .. الهدف بتاعنا اننا نتعلم كل البارمترز بتاعت الفانكشن ديه اللي بت induce a particular set Z_s for any x ... SGD باستخدام ال

Not really,
no target for hidden units...



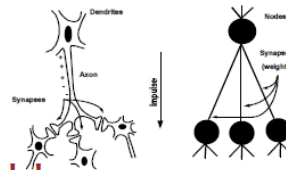
© Eric Xing @ CMU, 2006-2011

19

الدكتور بيقول ان يعني نخلي في دماغنا ال analogy ما بين البيولوجي و ال AI ..

From Biological to Artificial

The motivation for Artificial Neural Networks comes from biology...



Biological "Model"

- **Neuron:** an excitable cell
- **Synapse:** connection between neurons
- A neuron sends an **electrochemical pulse** along its **synapses** when a sufficient voltage change occurs
- **Biological Neural Network:** collection of neurons along some pathway through the brain

Biological "Computation"

- Neuron switching time: ~ 0.001 sec
- Number of neurons: $\sim 10^{10}$
- Connections per neuron: $\sim 10^4$
- Scene recognition time: ~ 0.1 sec

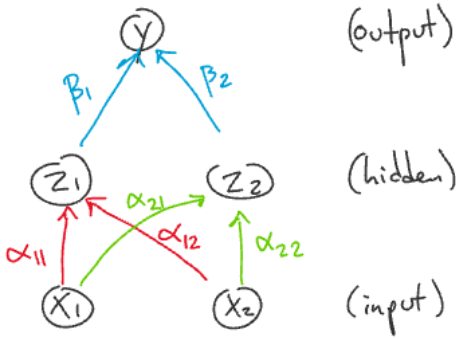
Artificial Model

- **Neuron:** node in a directed acyclic graph (DAG)
- **Weight:** multiplier on each edge
- **Activation Function:** nonlinear thresholding function, which allows a neuron to "fire" when the input value is sufficiently high
- **Artificial Neural Network:** collection of neurons into a DAG, which define some differentiable function

Artificial Computation

- Many neuron-like threshold switching units
- Many weighted interconnections among units
- Highly parallel, distributed processes

Ex#1: NN w/1 Hidden Layer and 2 Hidden Units



$x_i \in \mathbb{R}$
 $z_i \in (0,1)$ if σ is sigmoid
 $z_i \in \mathbb{R}$ more generally

Let σ be the activation function
 If σ is sigmoid: $\sigma(a) = \frac{1}{1 + \exp(-a)}$

$$z_1 = \sigma(\alpha_{11}x_1 + \alpha_{12}x_2 + \alpha_{10})$$

$$z_2 = \sigma(\alpha_{21}x_1 + \alpha_{22}x_2 + \alpha_{20})$$

$$y = \sigma(\beta_1 z_1 + \beta_2 z_2 + \beta_0)$$

$$= \sigma(\beta_1 \sigma(\alpha_{11}x_1 + \alpha_{12}x_2 + \alpha_{10}) + \beta_2 \sigma(\alpha_{21}x_1 + \alpha_{22}x_2 + \alpha_{20}) + \beta_0)$$

Each is a Logistic Regression Model Function

Don't forget the intercept terms

$$Pr[Y=1 | \vec{x}, \alpha, \beta]$$

⇒ predict using Bayes Optimal Classifier

decision function

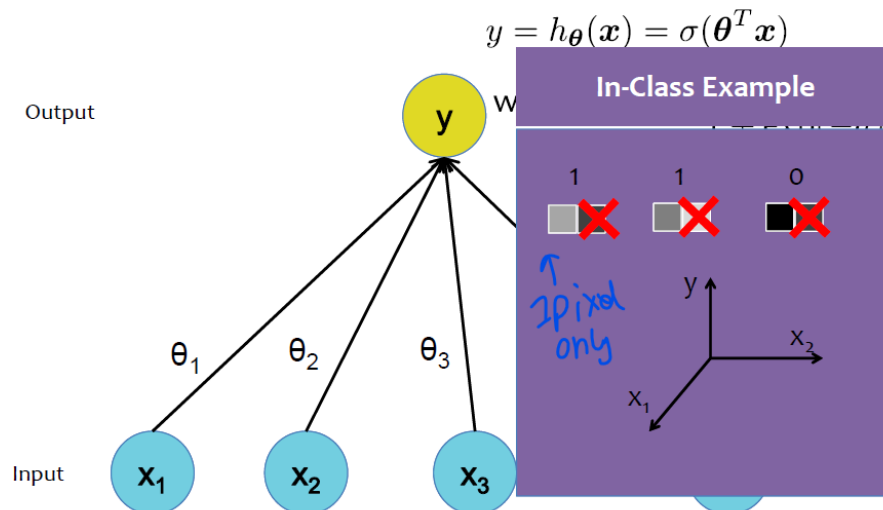
$$\hat{y} = h_{\alpha, \beta}(\vec{x}) = \begin{cases} 1 & \text{if } y > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

استخدما بيان تفعل
 Predictions و تقدر تستخدم ال SOD
 عشان تفعل ال Param. بت عنفا

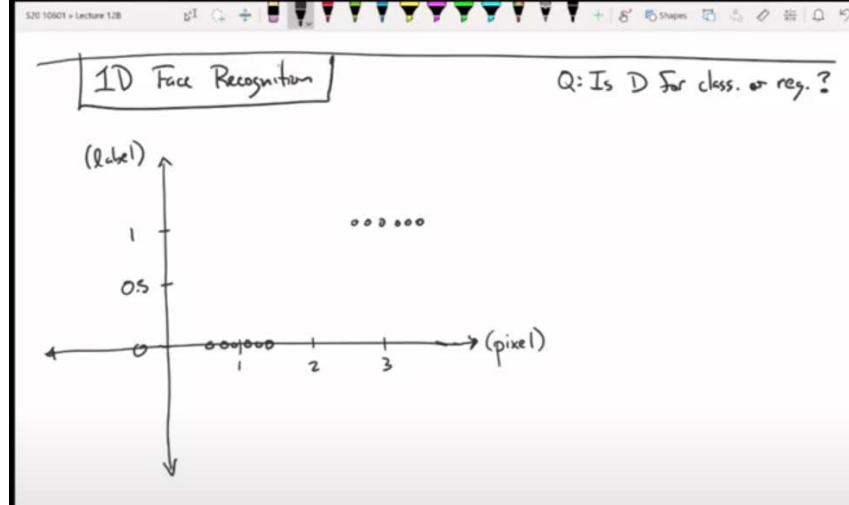
تعال نفكر في مثال بس الأول سريع كذا قلناه المره اللي فاتت لل facial recognition .. تعال نشوف ازاى ال facial recognition بيحصل ازاى ..

Decision Functions

Logistic Regression



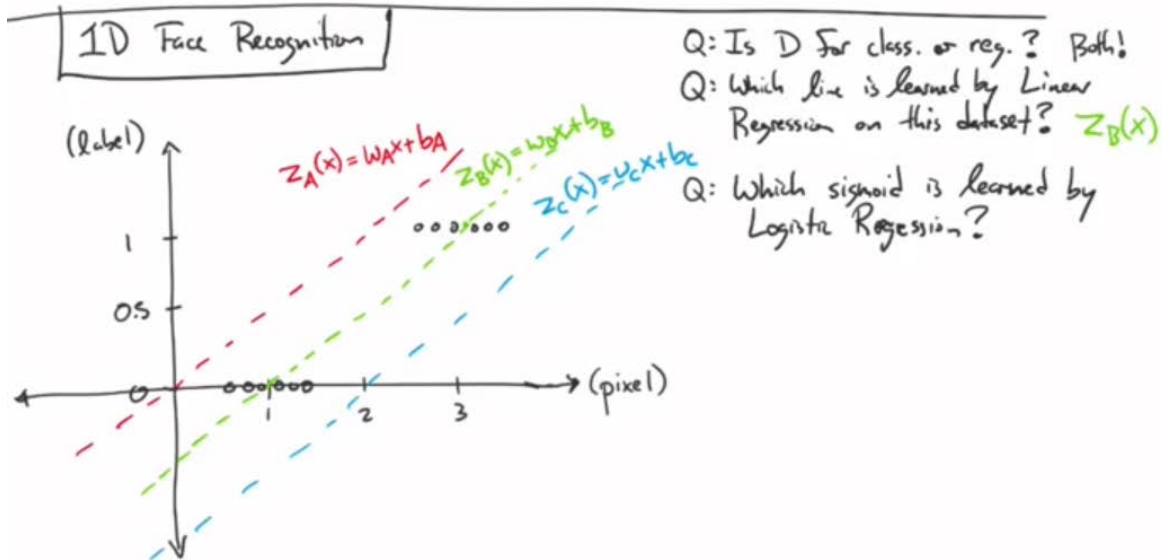
احنا مش عاوزين نفكر في الفانكشن نفسها .. احنا بس عاوزين نفكر في ال 1_D face recognition .. احنا هنقول ان احنا عندنا dataSet عايشه علي ال axis اللي في الصورة الجايه ديه



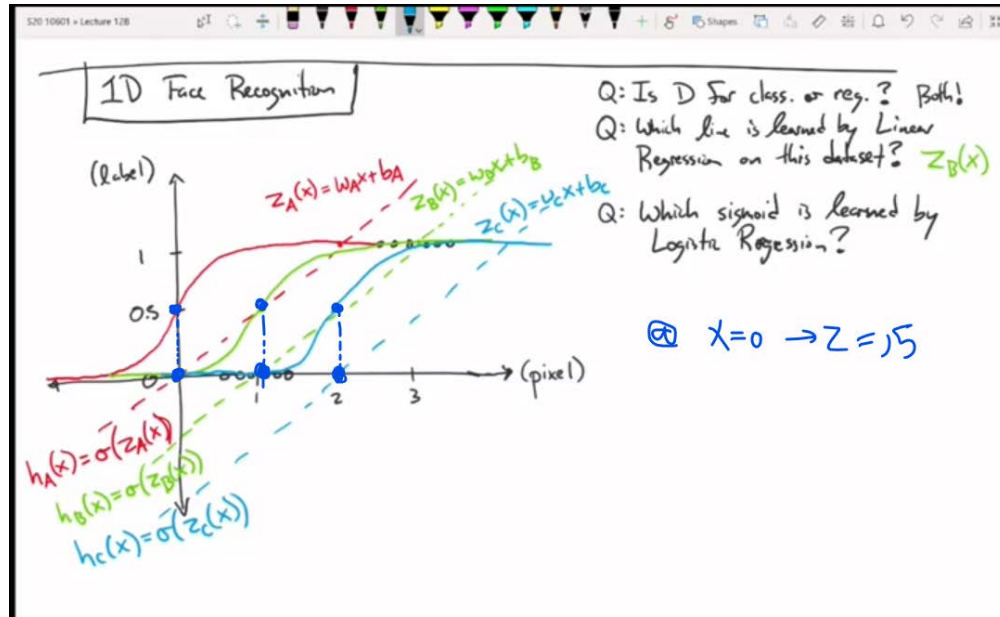
الدكتور سأل سؤال .. هل ديه مسأله كلاسيفيكشن ولا ريجريشن ... في حد رد وقالو كلاسيفيكشن .. الدكتور قالو صح فعلاً ... الإكس أكسيس هو البيكسلز و الواي هو الليبلز ... بس الدكتور كمل و قال هل حد عندو اجابه مختلفه ... حد قالو regression .. سألوه .. قالو عشان انت بتحاول ت minimize الاكتفيشن فانكشن وهي مش هتديلك يا صفر يا واحد .. فهنا ديه اصلا الرسمه اللي بنرسمها لل regression problem .. فالدكتور كمل كلام وقال انت فعلاً تقدر تفكر فيها كأنها regression داتاسيت .. وانت بتحاول تفيت فانكشن للداتا ديه .. "يحضر كومننت لدكتور pascal poupart من محاضره رقم 8

حساباتك .. دلوقت الهدف بتاعك انك تحل ال optimization problem عشان تلاقي ال W اللي بتقل ال objective function بتاعتك ... خيلنا بس نوضح حاجه الأول قبل ما نكمل .. التكنيك ده اسمو logistic regression بس فعلاً اللي انت بتحاول تعملو انك تعمل classification .. اللي هو لما جينا نتكلم في أول الكورس قلنا ان مئين ما بيكون عندنا كلاسيز تبقا categorical في الحاله ديه بتبقا classification انما لما بتحاول ت predict حاجه بتبقا numerical هي ديه الرجريشن .. هنا ال logistic regression هو تكنيك applied to a classification problem .. فده بيقا ليه سبب ورا الكلام ده .. انك لما بتفكر في اللي انت بتعملو فعلاً هو نوع من انواع الرجريشن عشان في الآخر انت بتحاول انك ت estimate the posterior $p(C|X)$ هي وديه نفسها اصلا رقم ما بين الصفر والواحد .. فده عشان كذا ده بيقا awkward شويه انو ريجريشن متطبق علي كلاسيفيكشن .. بس الحقيقه انك حولت الكلاسيفيكشن للريجريشن عن طريق انك بتقول انا هحاول predict البروبابلتي لكل كلاس و البروبابلتي مجرد رقم فبالتالي كذا بقت ريجريشن ..

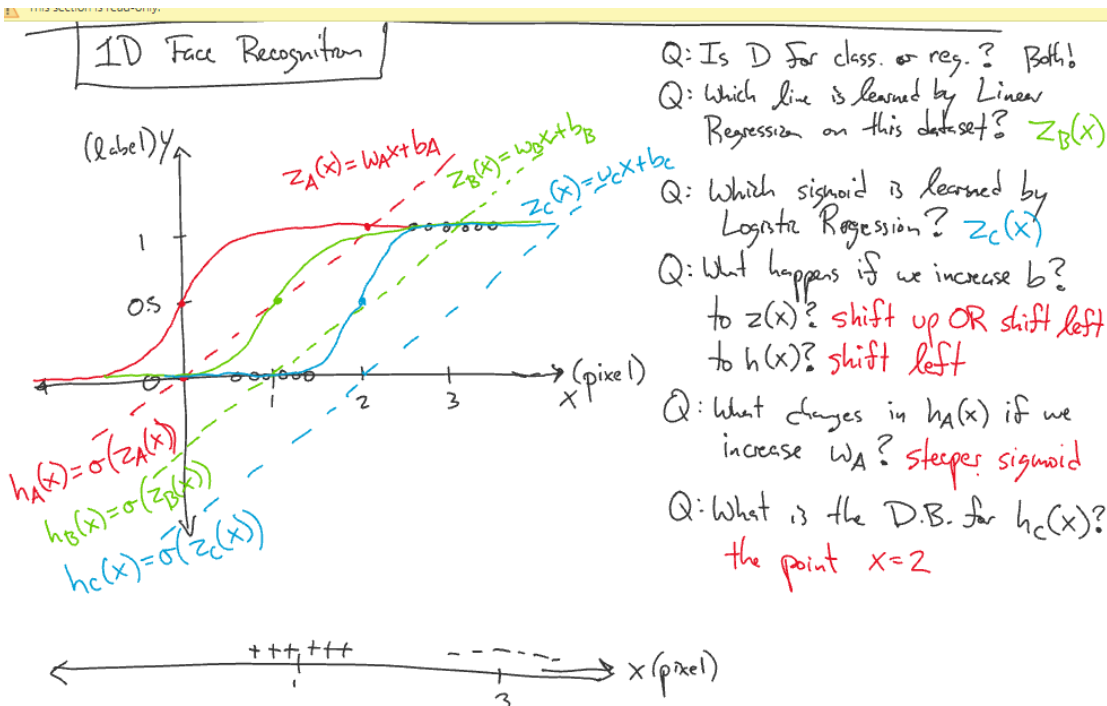
فالايجابه ع السؤال هتبقا الإثنين هي regression and classification .. الدكتور سأل سؤال ثاني .. أنهي خط هنتعلمو باستخدام ال linear regression علي الداتاسيت ديه ... الدكتور رسم 3 خطوط .. واختار الي في النص عشان هو اللي هيقلل ال mean squared error ..



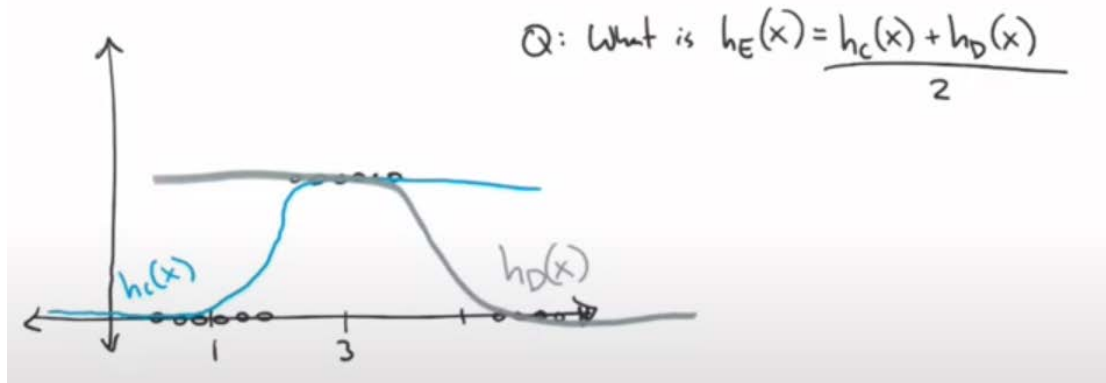
الدكتور سأل سؤال ثاني ... انه سجمويد هتبقا هي اللي موجوده واتعلمناها باستخدام ال logistic regression .. فقام قايل .. خد ال dot product اللي عندك لكل خط .. و دخلو علي سجمويد .. فاللي هو $h_A(x) = \text{sigmoid}(Z_A(x))$.. طيب ديه هيبقا شكلها عامل ازاي ...



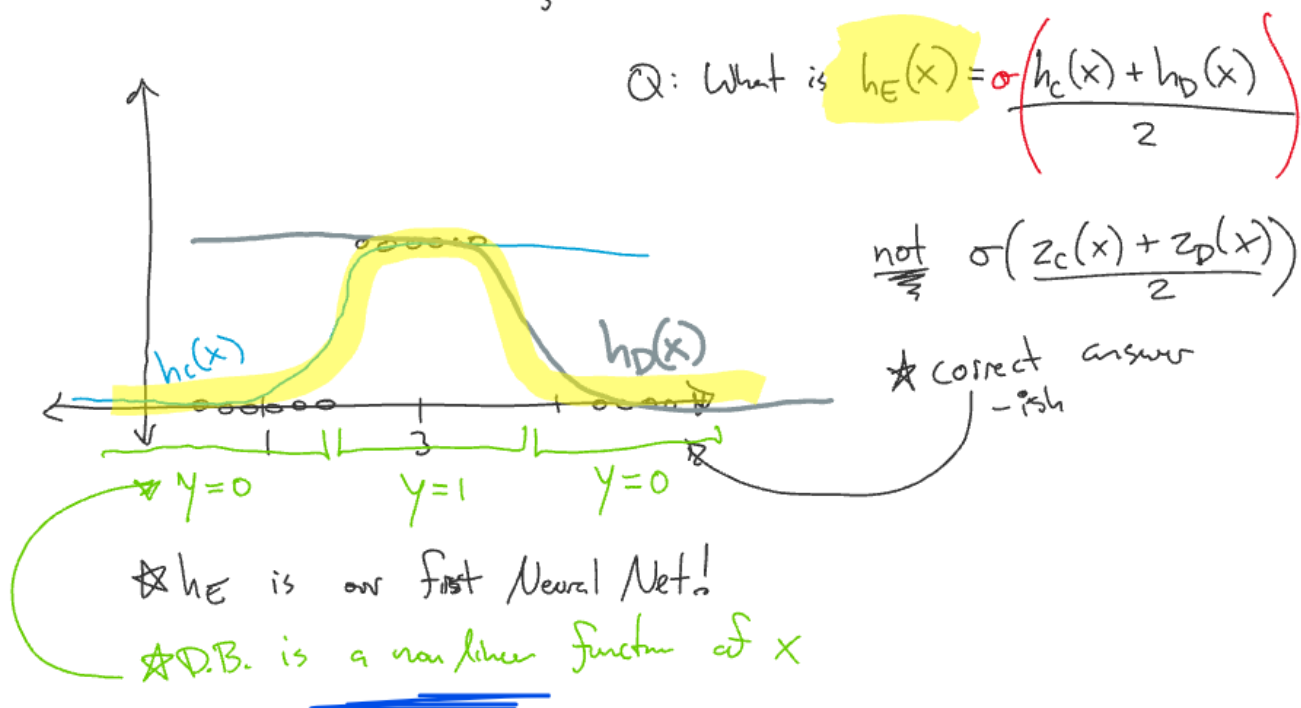
الدكتور بعد كذا راح ورسم الحوار ده كايو classification problem .. هيبقا عندك خط ... فيه موجب حوالين الواحد و سالب حوالين ال 3 .. وده اللي بنرسمو في اغلب ال classification انما في ال ريجريشن بتبقا زي اللي فوق كذا .. دلوقت عاوزين نسال علي ال behavior .. ايه اللي هحصل لو احنا زودنا ال b .. ايه اللي هحصل $z(x)$ و ال $h(x)$... الخط هي shift فوق ... معادله عادي ... في اجابه مكافئه للحوار ده .. انك تقول shift left ليه .. عشان وانت بتقول انك هتشف شمال و تخلي السلوب زي ماهو .. هي هي انك تخلي السلوب ثابت و تشف الخط لفوق .. طب بالنسبه لل $h(x)$.. ايه اللي هحصلها .. هتشف شمال برضو .. اللي هو بدل ما تقول شفت لفوق .. "كومننت: اللي هو م الآخر خد الكيرف بتاعك وشف اللوكيشن بتاع الصفر اللي علي الإكس أكسيس هتلاقه راح شمال فعلاً" .. طيب تعال نسال سؤال ثاني .. ايه اللي بيتغير في ال $h_A(x)$ لو احنا زودنا قيمة ال w_A .. اللي هو غيرنا السلوب بتاع الكيرف ... السجمويد هتبقا steeper .. في حد عبقرى والله .. ردع الدكتور في الشئ اللي قبل ديه وقالو انها هتبقا زي ال sign function .. وده حرفيا هو ال most steep اللي تقدر تجيبو .. أرخم سؤال في كل حاجه .. ايه هو ال D.B ال اللي مقابل ل $h_C(x)$... احنا هنا بنشتغل علي 1-D .. example .. تخيل لو احنا بنتكلم علي حاجه في ال 2-D .. ال D.B هيبقا عبارته عن line .. لو 3-D هيبقا DB .. فهنا ال DB هيبقا عبارته عن نقطه .. ايه هي النقطه .. هي $x=2$..



طيب احنا ليه بنخش في الحوارات ديه كلها .. انت دلوقت محتاج تفكر يعني ايه تحط 2 logistic regressions together



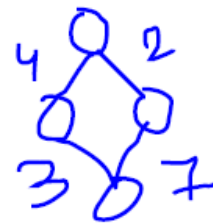
كدا بقا عندك حاجة شبه ال bell shaped curve ... احنا بنأخذ average 2 non-linear functions .. لو الدكتور كان قال هات الأفريديج بتاع ال z الاجابه مش زي منت قلت لل h انك هتعمل average of sigma(z_c + z_d) ... الدكتور بيقول ان الاجابه الصح هتبقا هي الكيرف اللي عندك ده .. و ده مش مفهوم عشان انت بتشتغل علي non-linear functions .. فالدكتور قام حاطط sigma حواليين ال h_c و ال h_d .. وده كدا هو أول neural network .. فهنا ال D.B بقا non-linear في الإكس .. بص ع الصورة هتلاقي في مناطق فيها صفر ومناطق فيها 1 ...



طبيب تعال نرجع لل face recognition .. هناخد دلوقة ال 2 pixels .. ازاي هنرسم الحوار ده .. احنا اشتغلنا علي واحد دابمينيشن ورحنا لفينا كتير .. الدكتور عمل اكتفتي في المحاضره رائاع .. اتفرج عليه

Neural Network Parameters

pr.mhcarve.019

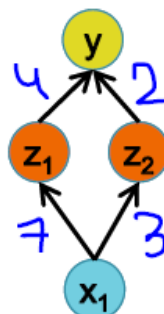


Question:

Suppose you are training a one-hidden layer neural network with sigmoid activations for binary classification.



True or False: There is a unique set of parameters that maximize the likelihood of the dataset above.



Answer:

$$A = \text{True}$$

$$B = \text{False}$$

newname
10/6/2021 12:43:49 AM

المفتاح في السؤال هنا .. هو كلمة unique .. انت ممكن الاجابه هتبقا غلط .. ليه مش unique .. انت ممكن بيضا عندك different weights بتديك نفس ال same بالازرق .. الدكتور بيقول انك لو افترضت ان الارقام علي الرسمه الاصليه كانت كذا .. انت ممكن تجيب نورال نتورك تانيه .. تعكس الارقام اللي موجوده .. وهتديك نفس ال decision boundary فالتالي اهو انت عندك different parameters بيدولك نفس ال likelihood .

المهم خذ بالك ان الأوبيجكتيف فانكشن اللي بتعملها optimization هتبقا non-convex ده عشان هيبقا عندك اجابتين مختلفين تماماً عن بعض .. maximize the likelihood of the dataset

29

دلوقت هنتكلم علي الأركتكشر بتاع النورال نتورك عشان حتي لو عندك basic NN عندك اختيارات كتيره هتعوّز تعملها

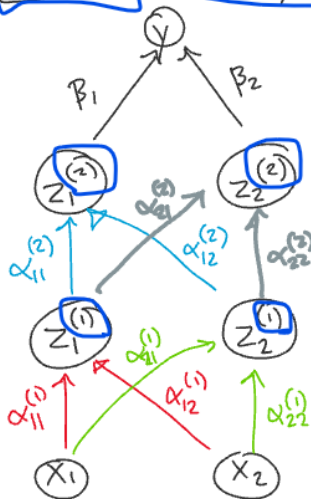
Neural Network Architectures

Even for a basic Neural Network, there are many design decisions to make:

1. # of hidden layers (depth)
2. # of units per hidden layer (width)
3. Type of activation function (nonlinearity)
4. Form of objective function

5. How to initialize parameters

Ex #2: NN w/ 2 Hidden Layers and 2 Hidden Units Each



$$z_1^{(1)} = \sigma(\alpha_{11}^{(1)} x_1 + \alpha_{12}^{(1)} x_2 + \alpha_{10}^{(1)})$$

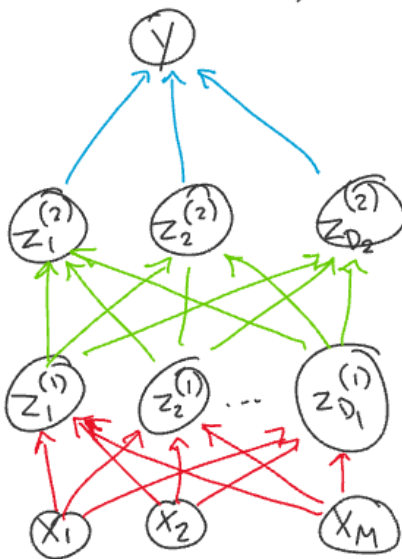
$$z_2^{(1)} = \sigma(\alpha_{21}^{(1)} x_1 + \alpha_{22}^{(1)} x_2 + \alpha_{20}^{(1)})$$

$$z_1^{(2)} = \sigma(\alpha_{11}^{(2)} z_1^{(1)} + \alpha_{12}^{(2)} z_2^{(1)} + \alpha_{10}^{(2)})$$

$$z_2^{(2)} = \sigma(\dots)$$

$$y = \sigma(\beta_1 z_1^{(2)} + \beta_2 z_2^{(2)} + \beta_0)$$

Ex #3: Arbitrary Feedward NN (Matrix Form)



parameters

$$\vec{\beta} \in \mathbb{R}^{D_2}$$

$$\beta_0 \in \mathbb{R}$$

$$\alpha^{(2)} \in \mathbb{R}^{D_1 \times D_2}$$

$$\vec{b}^{(2)} \in \mathbb{R}^{D_2}$$

$$\alpha^{(1)} \in \mathbb{R}^{M \times D_1}$$

$$\vec{b}^{(1)} \in \mathbb{R}^{D_1}$$

computation

$$y = \sigma(\vec{\beta}^T \vec{z}^{(2)} + \beta_0)$$

$$\vec{z}^{(2)} = \sigma((\alpha^{(2)})^T \vec{z}^{(1)} + \vec{b}^{(2)})$$

$$\vec{z}^{(1)} = \sigma(\underbrace{(\alpha^{(1)})^T}_{D_1 \times M} \underbrace{\vec{x}}_{M \times 1} + \underbrace{\vec{b}^{(1)}}_{D_1 \times 1})$$

σ applied elementwise to the vector $(\alpha^{(1)})^T \vec{x} + \vec{b}^{(1)}$

Fold in the Intercept Terms?

Assume $x_1 = 1, z_1^{(1)} = 1, z_1^{(2)} = 1$
Drop $\beta_0, \vec{b}^{(1)}, \vec{b}^{(2)}$

Caution: tricky to implement