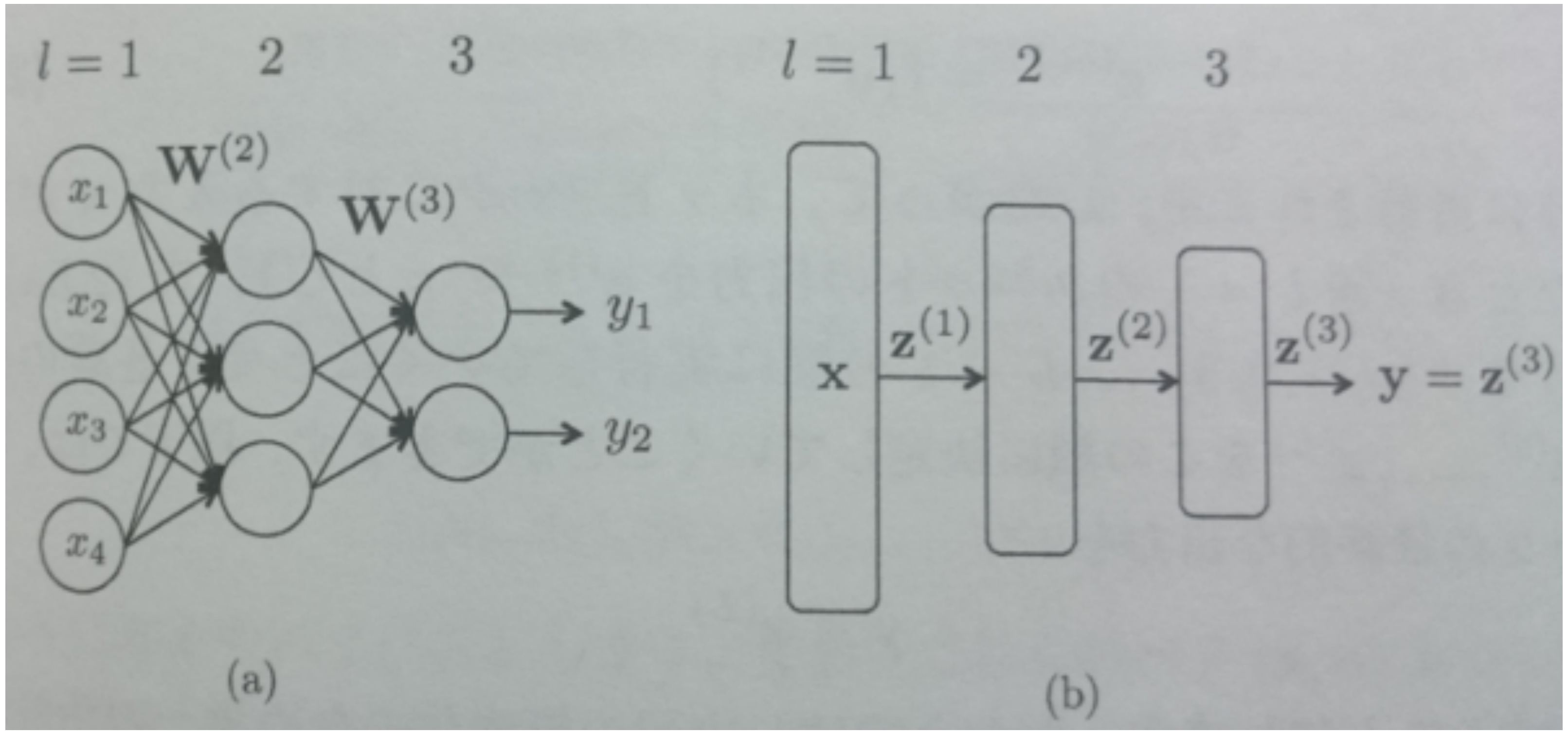
深層学習(MLP) 2章後半

順伝搬型ネットワーク

2.3多層ネットワーク

2層構造のネットワークでは各層をl=1,2,3で表し、l=1を入力層、l=2を中間層または隠れそう、l=3を出力層と呼ぶ



2.3多層ネットワーク

2層構造のネットワークの場合

l=1の層からの出力Xと入力~中間層間の重みをWとしたら

$$X = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}$$

$$W^{(2)T} = \begin{pmatrix} w_1 & w_2 & w_3 & w_4 \end{pmatrix}$$

中間層への入力u及び出力zは以下のように表せる

$$u^{(2)} = W^{(2)}X + b^{(2)}$$

$$z^{(2)} = f(u^{(2)})$$

2.3多層ネットワーク

任意の総数Lのネットワークにおいて、層l+1のユニットの出力は以下のように表せる

$$u^{(l+1)} = W^{(l+1)} z^{l} + b^{(l+1)}$$

$$z^{(l+1)} = f(u^{(l+1)})$$

2.4出力層の設計と誤差関数

2.4.1学習の枠組み

順伝搬型ネットワークはy(x;W)の関数で表すことができる。

N個の訓練用データは{(x1, d1),(x2,d2),,,,(xN,dN)}と表すことができ、関数y(x;W)の引数として渡した結果と実際の数値dとの差を測定することでパラメータWの正確さがわかる

この関数y(x;W)の結果と実際の値dとの結果を誤差関数と呼んでいる

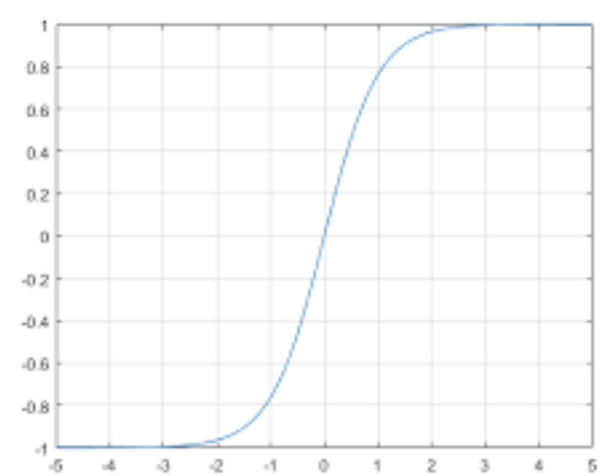
誤差関数: $(y(x_n;W) \approx d_n)$

2.4出力層の設計と誤差関数

2.4.2回帰

回帰とは？
主に出力に連続値をとる関数を対象に、訓練データをよく再現する関数を定めること
出力層の活性化関数には、その値域が目標とする関数と一致させる必要がある

値域が[-1:1]の場合
　→双曲線正接関数が適している
値域が[-∞:∞]の場合
　→恒等写像がよく使われる



双曲線正接関数

※恒等写像とは
その引数として用いたのと同じ値を常にそのまま返すような写像である。集合論の言葉で言えば、恒等写像は恒等関係である。

2.4出力層の設計と誤差関数

2.4.2回帰

回帰問題ではネットワークの出力y(xi)が訓練データの目標出力diに限りなく近くなるようにする必要があるが、それには「近さ」の尺度を決める必要がある。
一般的には二乗誤差が使われる

$$||d - y(x; w)||^2$$

全サンプルn=1,2,,,,Nについて加算して1/2したものがもっとも小さくなるようにパラメータWを決定する

$$E(w) = \frac{1}{2}\sum_{n=1}^{N} ||d_n - y(x_n; w)||^2$$

2.4出力層の設計と誤差関数

‣ 2.4.3二値分類

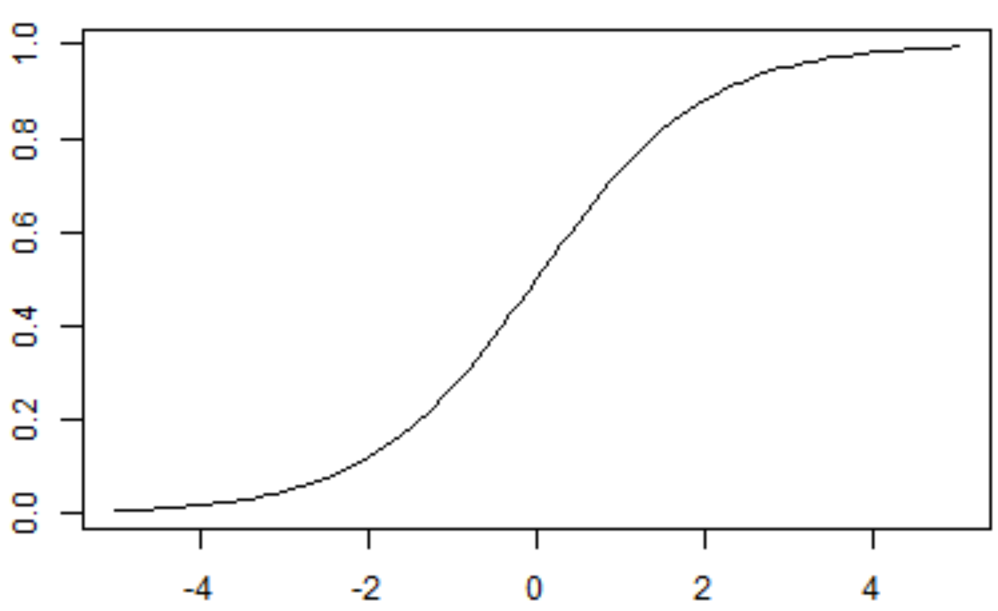
dを0または1の値をとるとしたとき(d ∊ {0,1})、d=1となる事後確率p(d=1 | x)をモデル化する方法を考える

→出力層はユニット1つだけ(変数dだけ)

活性化関数にはロジスティック関数を使い、0.5以上ならd=1, 未満なら0とする

$$y = \frac{1}{1+e^{-u}}$$

ロジスティック関数は[0,1]の値域をとるため確率を表すのに使うことができる



2.4出力層の設計と誤差関数

2.4.3二値分類

y=(x;w)を事後確立のモデルp(d=1|x) ≈ y(x;w)として、N個の訓練データ{(xn,dn) | n=1,,,N}を用いて事後分布p(d|x;w)が最もよく整合するようにパラメータWを決める。

ここで事後分布p(d|x; w)をd=1,0の事後分布を用いて以下のように表せる

$$p(d|x) = p(d=1|x)^d p(d=0|x)^{1-d}$$

d=0の場合はp(d=0|x)でd=1の場合はp(d=1|x)を1個の式でまとめている

このときパラメータwが訓練データをよく表しているのであればp(d|x)の値は大きくなる

2.4出力層の設計と誤差関数

2.4.3二値分類

訓練データに対するwの最もらしさ(尤度)は誤差を掛け合わせる形で与えられる

$$L(w) \equiv \prod_{n=1}^{N} p(d_n | x_n; w) = \prod_{n=1}^{N} \{y(x_n; w)\}^{d_n} \{1 - y(x_n : w)\}^{1-d_n}$$

上記の式のwを最適化することは、対数関数(この場合は対数尤度)を最適化することと同じである。

上記式ではL(w)が大きくなるように最適化し、下記E(w)では小さくなる(誤差が)ように最適かする

$$E(w) = -\sum_{n=1}^{N} [d_n \log y(x_n; w) + (1 - d_n) \log\{1 - y(x_n; w)\}]$$

2.4出力層の設計と誤差関数

2.4.3二値分類

活性化関数にロジスティック関数を使った解釈について

事後確立p(d=1|x)は以下のようにかける

$$p(d=1|x) = \frac{p(x,d=1)}{p(x,d=0)+p(x,d=1)}$$

上記の事後確立は下記uのロジスティック関数に一致する

$$u \equiv \log \frac{p(x,d=1)}{p(x,d=0)}$$

$$y = \frac{1}{1+e^{-u}}$$

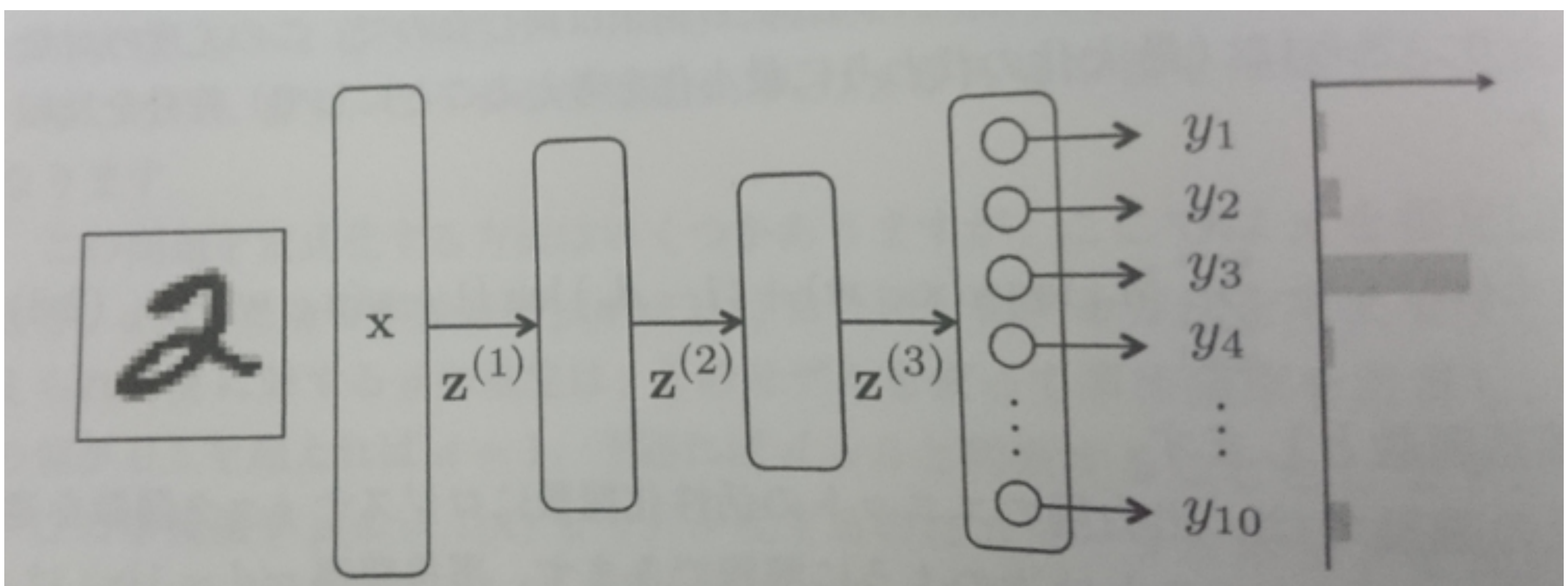
2.4出力層の設計と誤差関数

2.4.4多クラス分類

例えば32×32の画像に書かれた0〜9の数字を判定する場合、

入力は各ピクセルの画素値1024個で、出力には0~9のそれぞれに数値に一致する確率

となる(MNISTと言って機械学習とかの入門とかでよく使われている)



2.4出力層の設計と誤差関数

2.4.4多クラス分類

出力層l=Lの各ユニットk(=1,2,,K)の総入力は1つ下の層l=L-1の出力を元に

uk^(L) (=W^(L)z^(L-1) + b^(L))と与えられ、出力層のk番目のユニットの出力には

ソフトマックス関数を使い以下のように表せられる。

$$y_k \equiv z_k^{(L)} = \frac{e^{u_k^{(L)}}}{\sum_{j=1}^{K} e^{u_j^{(L)}}}$$

ソフトマックス関数により出力層にある各ユニットの出力の合計が1になるようにしている

多クラス分類ではクラスがK個あるとき、y0,y1,,,yKのうち最も大きいkのクラスに分類するようにします。

2.4出力層の設計と誤差関数

2.4.4多クラス分類

多クラス分類の出力層の訓練データについて、例えば0~9の数値を分類する多クラス分離で2の訓練データの出力層は以下のようになる

d=[0 0 1 0 0 0 0 0 0 0]

これより、事後分布は以下で表すことができる

$$p(d|x) = \prod_{k=1}^{K} p(C_i|x)^{d_k}$$

d=[0 0 1 0 0 0 0 0 0 0]の時はp(d|x) = 1 * 1 * p(c2|x) * 1 * 1 * 1 * 1 * 1 * 1 * 1 = p(c2|x)

訓練データがN個ある場合のwの尤度は以下のように導出できる

$$L(w) = \prod_{n=1}^{N} p(d_n|x_n; w) = \prod_{n=1}^{N} \prod_{k=1}^{K} (y_k(x; w))^{d_{nk}}$$

2.4出力層の設計と誤差関数

2.4.4多クラス分類

尤度が求められたので、以下の対数尤度を誤差関数として使用できる。

尤度: $$L(w) = \prod_{n=1}^{N} p(d_n | x_n; w) = \prod_{n=1}^{N} \prod_{k=1}^{K} (y_k(x; w))^{d_{nk}}$$

対数尤度: $$E(w) = -\sum_{n=1}^{N} \sum_{k=1}^{K} d_{nk} \log y_k(x_n; w)$$

この対数尤度は交差エントロピーと呼ばれている

2.4出力層の設計と誤差関数

2.4.4多クラス分類

活性化関数にソフトマックス関数を使っているけど
ckの事後確立は以下のように表せる

$$p(c_k|x) = \frac{p(x, c_k)}{\sum_{j=1}^{K} p(x, c_j)}$$

ここでuk ≡ log(p(x, ck)) → p(x, ck) ≡ exp^(uk)とすると

$$p(c_k|x) = \frac{e^{u_k}}{\sum_{j=1}^{K} e^{u_j}}$$

となり、ソフトマックス関数に一致することがわかる