

ÚLOHA 6

Vícevrstvý perceptron

Zadáno na cvičení: 7
Mezní termín: 22.11. 2017
Maximální počet bodů: 20
Nepovinná úloha

Zadání

Stáhněte si archiv *perceptron.zip* ze stránky *Neuronové sítě*. Archiv obsahuje tyto soubory:

- *checkNNGradients.m* – Porovná analyticky vypočítané gradienty s numerickými odhady.
- *computeNumericalGradient.m* – Vypočítá numerický odhad gradientu.
- *data.mat* – vstupní data v binárním formátu MATLABu.
- *debugInitializeWeights.m* – Inicializuje váhy na pevné hodnoty (pro ladící účely).
- *displayData.m* – vizualizace dat
- *fmincg.m* – pokročilá optimalizační funkce
- *getMultilayerPerceptron.m* – pro vytvoření vícevrstvého perceptronu.
- *gradientDescent.m** – gradientní sestup
- *nnCost.m*** – cenová funkce perceptronu
- *nnPredict.m*** – hypotéza vícevrstvého perceptronu
- *perceptronExample.m* – spouštěcí skript úlohy
- *randInitializeWeights.m*** – náhodná inicializace vah
- *sigmoid.m** – logistická funkce (sigmoida)
- *sigmoidGradient.m*** – funkce pro výpočet gradientu sigmoidy
- *weights.mat* – natrénovaný model (optimální váhy).

Soubory označené* máte naprogramované z předchozích úloh.

Soubory označené** budete doplňovat.

1 Klasifikace do více tříd

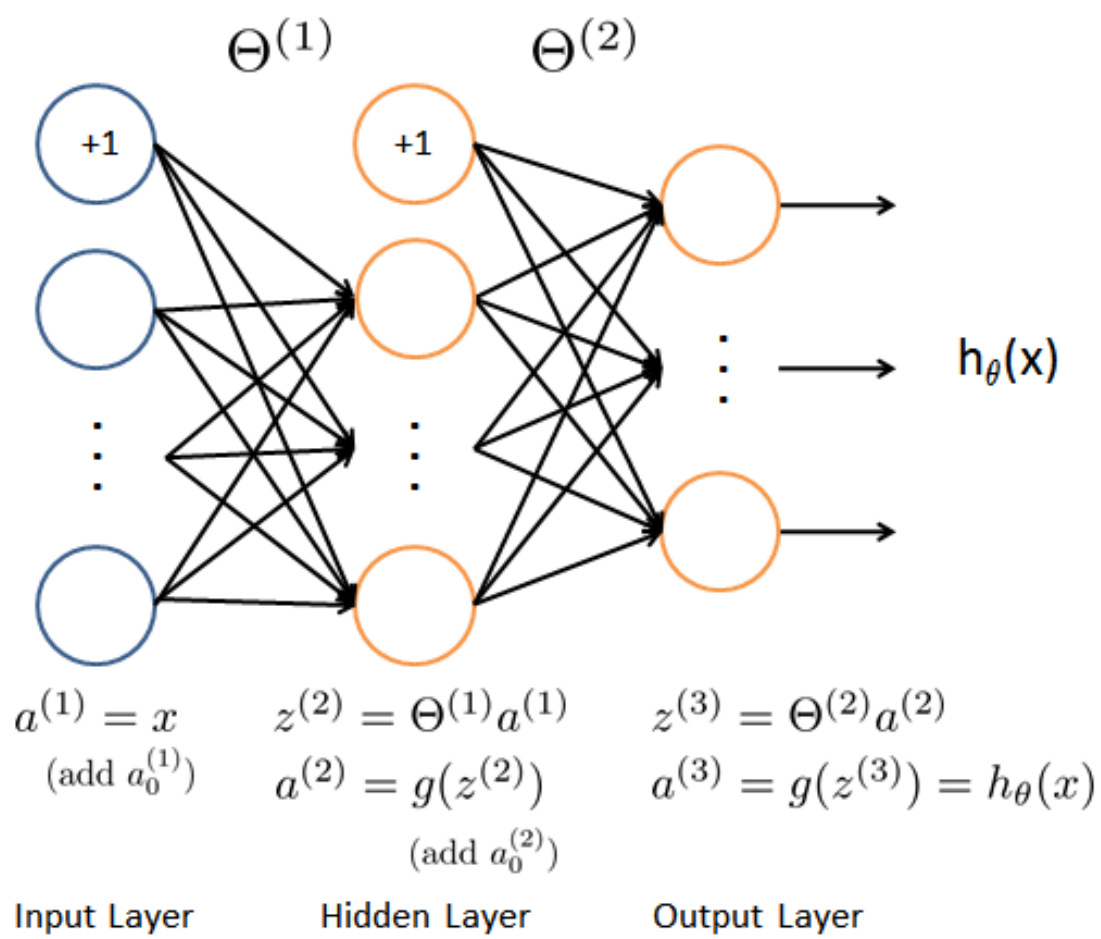
V této části použijete vícevrstvý perceptron k rozpoznávání ručně psaných číslic (stejná úloha jako v předchozí úloze na logistickou regresi). Čísla jsou uložena jako bitmapa 20x20. Přímo v této surově podobě vstupují do klasifikátoru a je na perceptronu, aby si z dat vytvořil rozumné příznaky. Vstupní vrstva má tedy 400 neuronů (+ absolutní člen), skrytá vrstva má 25 neuronů (tuto hodnotu můžete zkusit změnit, jedná se o počet abstraktních vzorů, které se síť bude snažit naučit). Výstupní vrstva má 10 neuronů (protože klasifikujeme do deseti tříd).



Obrázek 1: Část trénovacích dat

Úkoly

1. **Naprogramujte náhodnou inicializaci vah.** soubor *randInitializeWeights.m* (2 body)
2. **Naprogramujte dopředné šíření, vypočítejte výstup a chybu.** Budete upravovat funkce *nnPredict.m* a *nnCost.m*. (5 bodů)
3. **Naprogramujte výpočet gradientu chyby podle všech vah.** Budete upravovat soubor *nnCost.m* a *sigmoidGradient.m*. (5 bodů)
4. **Do cenové funkce a jejího gradientu implementujte regularizaci.** Budete upravovat soubor *nnCost.m*. (3 body)
5. **Naprogramujte chybu a její gradient pomocí maticových operací** (bez cyklů). Doporučuji nejdříve spočítat backprop v cyklu přes data a až to odladíte, pokuste se výpočet přepsat maticově. Budete upravovat soubor *nnCost.m*. (5 bodů navíc)



Obrázek 2: Schéma vícevrstvého perceptronu