

Linjär och Kombinatorisk optimering

Fredrik Kahl

Matematikcentrum

Föreläsning 13: Neurala nätverk

- Introduktion
- Perceptron
- Flera lager
- Support Vector Machines
- Hopfield
- Boltzmannmaskinen

Introduktion

- Ett nytt beräkningsätt
- Parallellitet ger snabbhet
- Inläring
- Optimering

Historik

Starten

- MacCulloch & Pitts, 1943
- Wiener & von Neumann
- Hebb, 1949

Första lyckade försöket

- Rosenblatt, 58

Nedgång

- Minsky-Papert kritiken, 69

Andra vågen (80- och 90-talet)

- Hopfield
- DARPA
- Journal Neural Networks, Neural Computation, IEEE Trans on N N
- Vapnik och Support Vector Machines (SVM)

Biologi

Hjärnan

$\approx 10^{11}$ neuroner,

$\approx 10^4$ synapser per neuron,

$\approx 10^{16}$ operationer/sekund,

$\approx 10^{-16}$ Joule/operation,

Mikroprocessor

$\approx 10^8$ operationer/sekund,

$\approx 10^{-8}$ Joule/operation,

Perceptronen

Rosenblatts perceptron 1958

Inspiration från mönsterigenkänning

$$y = f\left(\sum_{j=1}^n w_j u_j + b\right)$$

där f är stegfunktionen.

Minsky-papert kritiken

Problemet med xor

Begränsning: Linjärt separabla klasser

Tvålayersperceptronen

Tvålayersperceptronen

Löser XOR-problemet.

Multilager perceptroner

- Framåtkoppling
- Approximerar icke-linjära funktioner.
- Teoretiskt stöd för att många funktioner går att approximera på detta sätt.
- Ett gömt lager räcker i princip.
- Empiriskt har det visat sig att flera lager kan vara användbara.
- Föreslogs redan av Rosenblatt
- Hur många dolda noder ska man ha?
- Hur ska man träna ett nät?

Support Vector Machines

- Liknande enlayersperceptronen, men indata x är först avbildat till ett annat vektorrum, kallat 'feature space'

$$\Phi : x \longrightarrow \Phi(x)$$

- Φ är vanligtvis ickelinjär och beror på applikation
- Dimension på "feature space" är vanligtvis mycket högre än dimensionen på "input space", kan t o m avbildas till \mathbf{R}^∞ .

Inläring

- Handledd inläring. Man lär nätet ett specifikt beteende.
- Lärarlös inläring. Nätet får själv strukturera hitta lämpliga

Inspiration från biologin.

Donald Hebb (Kanadensisk psykolog) The Organization of Behaviour, Wiley, 1949.

Hebbs inlärningsregel (med lärare)

Om en synaps orsakar en neuron att ge önskat utsignal så ökar man synapsstyrkan.

$$\Delta w = \gamma xy$$

Inläring - Ett optimeringsproblem

Betrakta inläringen som ett optimeringsproblem. Det gäller att ställa in alla variabler så att man får ett önskat beteende.

Använd bara deriverbara funktioner.

Använd klassisk optimeringsteori för att hitta optimum.

Försök omvandla dessa ideer till snabba distribuerade algoritmer som fungerar lokalt.

Om neuronmodellen är $y = F_w(x)$ och vi har ett antal stimuli som vi vet facit på, hitta parametrarna genom att lösa

$$\min_w \sum_{k=1}^N |y_i - F_w(x_i)|^2$$

Steepest descent - Backpropagation

Exempel: Ett lager.

Vi vill att nätet ska svara y på stimuli x .

Välj vikter w genom att lösa

$$\min_w V(w) = \min_w \frac{1}{2} (y - f(w^T x))^2$$

Använd Steepest descent

$$\Delta w = -\gamma \frac{\partial V}{\partial w} = -\gamma e f'(w^T x) x$$

Nästan som Hebb's regel.

Liknande för flera lager.

Hopfieldnät

Godtyckliga neuronkopplingar

$$C_i \frac{du_i}{dt} = \sum_j T_{ij} f_j(u_j) - u_i/R_i + I_i$$

- Neuron - ManyInSingleOut
- Neglect pulse nature
- Use simple dynamics
- Why is dynamics used in a net with feedback?
- Different non-linearities

Hopfieldnät för optimering

Ett Hopfield nät strävar efter att minimera sin energifunktion.

$$E = -\frac{1}{2} \sum_{i,j} T_{ij} V_i V_j + \sum_i \frac{1}{R_i} \int_0^{V_i} f_i^{-1}(V) dV - \sum_i I_i V_i$$

Derivera och visa att $dE/dt \leq 0$.

Idé: Konstruera ett hopfieldnät vars energifunktion motsvarar målfunktionen f som vi vill minimera.

Hopfield and Tank, 1986.

Nätet blir då en parallell implementation för att söka lokala minima.

Exempel: Handelsresandeproblemet.

Exempel med explicita vikter.

Boltzmannmaskinen

Boltzmannmaskinen är ungefär som ett Hopfield nät med stokastisk neuronavfyrning.

Definition Boltzmannmaskin.

Ett antal tillstånd (noder) V_i som kan vara antingen på eller av, dvs $V_i = 0$ eller 1.

Bågar med vikt s_{ij} som stimulerar $s_{ij} > 0$ eller inhiberar $s_{ij} < 0$ ett nodpar.

Konsensusfunktion.

$$C = \sum_{i,j} s_{ij} V_i V_j$$

Boltzmannmaskinen maximerar konsensusfunktionen, antingen sekventiellt eller parallellt.

Trick: Konstruera en Boltzmannmaskin vars konsensusfunktion är lika med målfunktionen.

Boltzmannmaskinen försöker då hitta det optimum ungefär som simulerad stelning, men eventuellt med parallell implementation.

Repetition - Några typer av nät

- Flerlayersnät
- Support Vector Machines
- Hopfieldnät
- Boltzmannmaskinen

Repetition - Kopplingar till optimering

- Inlärning - optimering
- Hopfieldnät och Boltzmannmaskinen kan användas som parallella implementationer av optimeringsalgoritmer.