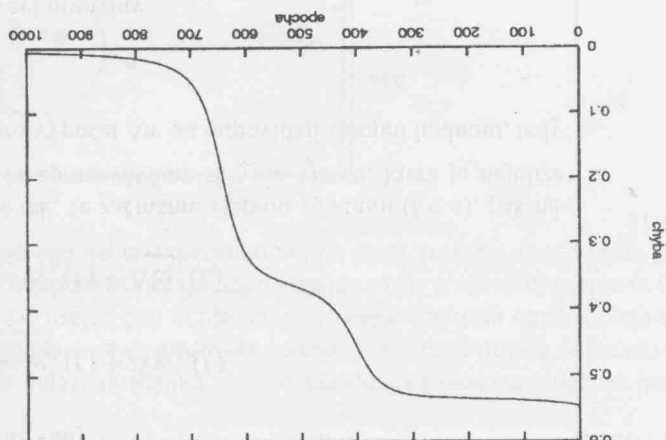


Bohužel existují případy, kdy se nám nepodaří síť naučit s dostatečně malou chybou. V takovém případě dosáhne energie určité hodnoty a nadále přestane klesat. Tomuto jevu říkáme uváznutí v lokálním minimu.

Při odvození algoritmu zpětného šíření jsme si řekli, že hledání požadované konfigurace vah můžeme přirovnat k putování po horském masivu. Správnou cestu hledáme podle toho, jak se terén svažuje, abychom se dostali do údolí. Problémem je, že jak ve váhovém prostoru, tak ostatně i v horách, je řada prohlubní, či malých údolí, která nejsou tím hledaným nejnižším položeným minimem. Pokud bychom při cestování do takového údolí (lokálního minima) dostali, nevěděli bychom přesně kudy se dát, protože každý krok by znamenal vzestup energie. Pokud se algoritmus zpětného šíření dostane do lokálního minima pokles energie se zastaví (viz Obr. 5.12).

Obr. 5.11 Typický průběh globální chyby během učení



Algoritmus zpětného šíření

Krok 1. Inicializace

Všechny váhy v síti nastavíme náhodně na hodnoty v doporučeném rozsahu $<0.3, 0.3>$.

Krok 2. Předložení vzoru

Vybereme vzor z trénovací množiny a přiřadíme na vstupy sítě. Dále po vrstvách směrem od vstupu k výstupům sítě počítáme výstupy jednotlivých neuronů dle vztahů:

$$y = S\left(\sum_{i=1}^n w_i x_i + \Theta\right) \text{ a } S(\phi) = \frac{1}{1 + e^{-\phi}}.$$

Krok 3. Srovnání

V kroku srovnání nejprve vypočítáme energii podle vztahu $E = \frac{1}{2} \sum_{i=1}^n (y_i - d_i)^2$, která bude využita jako přírůstek k celkové energii počítané přes všechny testovací vzory. Dále vypočítáme chybu pro výstupní vrstvu:

$$\delta_o^i = (d_i - y_o^i) y_o^i (1 - y_o^i).$$

Krok 4. Zpětné šíření chyby a modifikace vah

Pro všechny neurony ve vrstvě vypočteme:

$$\Delta w_{ij}^l(t) = \eta \delta_o^i(t) y_{j-1}^l(t) + \alpha \Delta w_{ij}^l(t-1),$$

$$\Delta \Theta_i^l(t) = \eta \delta_o^i(t) + \alpha \Delta \Theta_i^l(t-1).$$