

NEURONHÁLÓS HANGTÖMÖRÍTÉS

Áfra Attila Tamás

Tartalom

- Bevezetés
- Prediktív kódolás
- Neuronhálós prediktív modell
- Eredmények
- Források

Bevezetés

Digitális hanghullámok

Pulzus kód moduláció

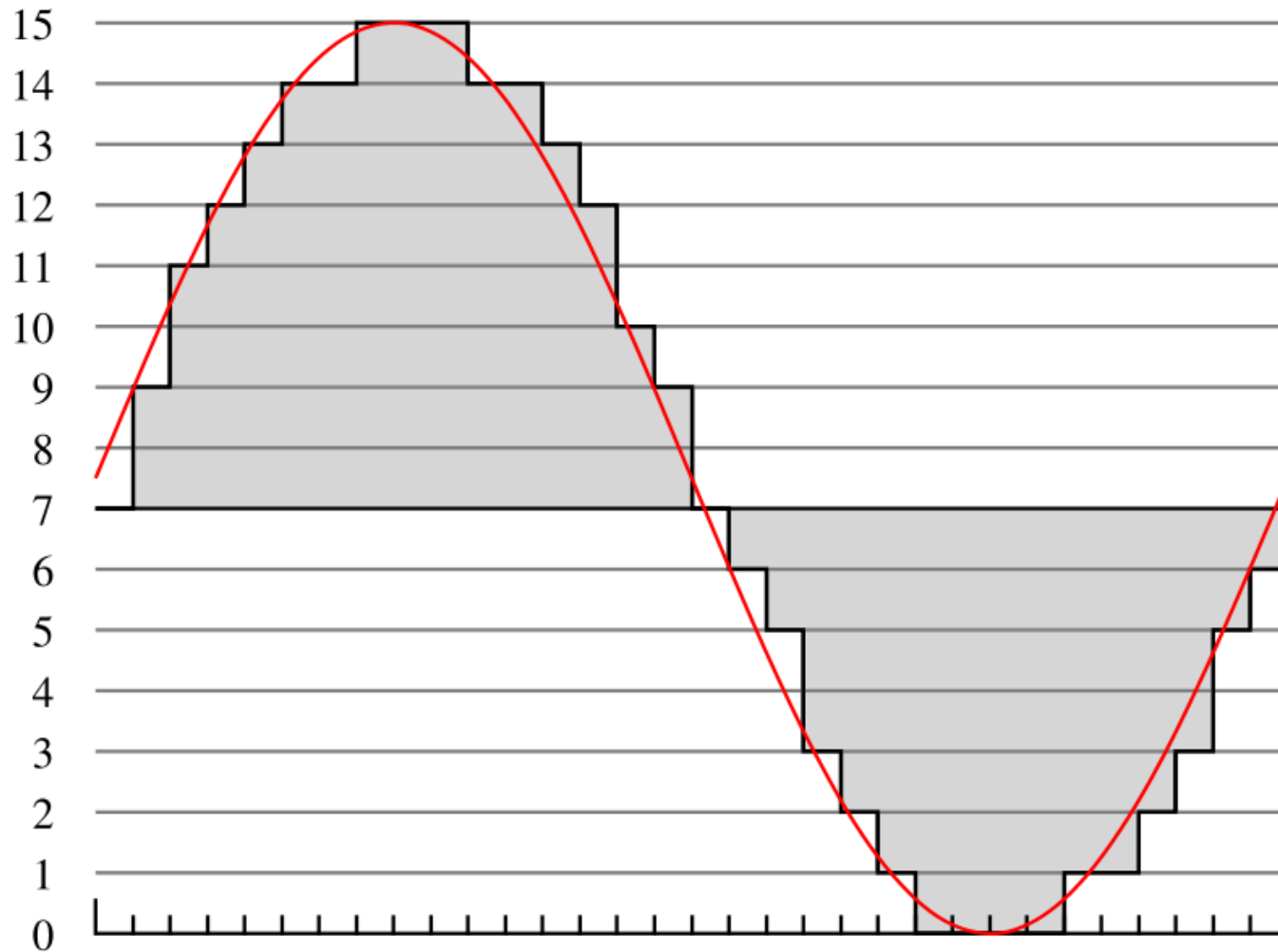
Hangtömörítés

Veszteségmentes hangtömörítés

Digitális hanghullámok

- Az analóg hanghullámok folytonosak.
- Digitalizálás során a jelet diszkrét elemekre kell bontani, vagyis mintákat kell venni egyenlő időközönként, ugyanakkor a minták is véges precizitásúak kell legyenek.
- A legelterjedtebb digitalizálási módszer a **PCM** (***P**ulse **C**ode **M**odulation* = pulzus kód moduláció).
- Az ilyen módon digitalizált jelet két paraméterrel lehet jellemezni: **bitmélység** (a minták pontossága) és **mintavételezési frekvencia** (felbontás).

Pulzus kód moduláció (4 bit)



Pulzus kód moduláció

- Ahhoz, hogy jó minőségű digitális hanghullámot kapjunk, minimum 16 bites mintapontosságra és 44.1 kHz-es mintavételezésre van szükség (CD minőség).
- Általában stereo a hangforrás, ezért egy helyett két mintára van szükség időegységenként.
- Manapság egyre elterjedtebb a 24 bites pontosság, 96-192 kHz-es mintavételezés és 6/8 csatorna (pl. DVD-Audio).

Hangtömörítés

- Miért van szükség hangtömörítésre?
- **1 órányi** 24 bites, 96 kHz-es, 6 csatornás hangadat mérete kb. **6 GB** !
- A legelterjedtebb CD minőségű hang veszteséges tömörítésére alkalmas hangformátum az MP3.
- Manapság sokkal nagyobb az adathordozók tárolási kapacitása, mint 13 éve, az MP3 megjelenésekor, ezért nincs már szükség olyan nagy mértékű, veszteséges tömörítésre.
- A hangminőség javításán van a hangsúly.

Veszteségmentes hangtömörítés

- Professzionális stúdió minőség eléréséhez elengedhetetlen, hogy a tömörítés veszteségmentes legyen, vagyis a kitömörített adat tökéletesen megegyezzen az eredetivel.
- A tárolási kapacitással együtt a hangadatok paraméterei is egyre jobban növekednek, ezért fontos, hogy a tömörítés hatékony is legyen.
- Általános célú tömörítési algoritmusokkal csak nagyon kis mértékben lehet tömöríteni hangot, ezért speciális módszerekre van szükség.
- Jelen pillanatban a legelterjedtebb veszteségmentes hangtömörítési formátum a FLAC.

Prediktív kódolás

Modell

Frame

Reziduális jel

Kódolás

FLAC

Modell

- A hangminták nem teljesen függetlenek egymástól.
- Ezt használja ki a **prediktív kódolás**.
- A soron következő mintát általában kicsi hibával meg lehet jósolni az előzők alapján.
- A hatékonyság és sebesség érdekében mindig csak az előző **N** mintát, a **kontextust**, kell figyelembe venni (**N** a **kontextushossz**).
- A jóslást egy olyan függvény végzi el, amelynek **N** paramétere van és a becsült értéket adja vissza.
- Ez a függvény képezi a tömörítési **modellt**.

Modell

- A modell egyik fontos jellemzője, hogy kevés memóriát igényeljen, így a tömörített fájlban el lehet tárolni anélkül, hogy jelentősen megnövelné a méretet.
- Ezért elég, ha a modell csak a tömörítendő hanghullám esetében hatékony.
- Ez csak abban az esetben érhető el, ha alkalmazkodik az illető hangadatokhoz.
- Tehát a prediktív modell **adaptív** kell legyen.

Frame

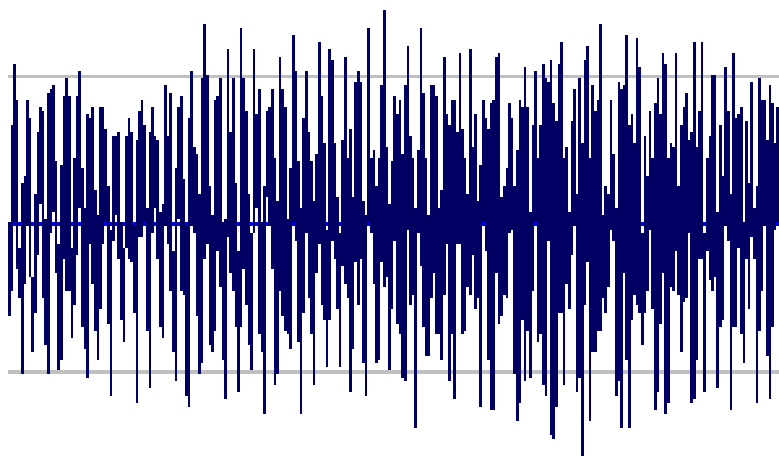
- A hangfájl könnyebb kezelhetősége és a modell nagyobb hatékonysága érdekében a hanghullámot kisebb blokkokra, ú.n. **frame**-ekre kell bontani.
- Minden *frame* rendelkezhet saját modellel, így a jóslás pontosabb lehet, mivel az illető modell jobban tud alkalmazkodni rövidebb blokkokhoz.
- A *frame* méretének meghatározása nem könnyű feladat, mivel a tömörítés kevésbé lesz hatékony, ha a méret túl kicsi vagy túl nagy (ekkor a seek művelet, vagyis tetszőleges pozícióra való ugrás, lassú).
- Általában egy *frame* pár ezer mintából áll.

Reziduális jel

- Mivel a modell nem tökéletes, ezért szinte mindig lesz eltérés az eredeti minta és a jósolt minta között.
- Ez az eltérés a **predikciós hiba**.
- Ahhoz, hogy visszacapjunk az eredeti jelet, elég csak a hibaértékeket eltárolni.
- A hibaértékek összessége képezi a **reziduális jelet**, ami az eredeti és a jósolt jel különbsége.
- A reziduális jelben sokkal nagyobb a redundancia mint az eredetiben, ezért annak hatékony kódolása tömörítéshez vezet.

Reziduális jel

Eredeti jel



Reziduális jel



Kódolás

- Mivel a hibaértékek statisztikai eloszlása általában nem egyenletes, vagyis egyes értékek gyakrabban fordulnak elő, mint mások, ezért lehet **statisztikai kódolást** alkalmazni.
- Több ilyen kódolási algoritmus is létezik: Huffman, Rice, aritmetikai kódolás, stb.
- Hang- és képtömörítésnél a Rice (a Huffman egy speciális esete) kódolás a legelterjedtebb, amely egyben a leggyorsabb is, viszont csak ebben a doméniumban hatékony, más típusú adatok tömörítésére nem alkalmas.

Kódolás

- A kódoláson nem lehet jelentősen javítani a jelenlegi módszerekhez képest, mivel az eddig ismert algoritmusokkal már majdnem elérhető az optimális tömörítés.
- Ezért egy prediktív kódolás alapú tömörítő ereje a modellben rejlik.
- Az ilyen tömörítés nagyon rugalmas is, mivel egy kódolót össze lehet kapcsolni több, egymástól akár radikálisan különböző, modellel is.

FLAC

- A FLAC (**F**ree **L**ossless **A**udio **C**odec) egy lineáris prediktív modellt használó tömörítő / fájlformátum.
- A nagyobb hatékonyság érdekében a kódolás előtt optimalizálja a reziduális jelet.
- Az egyik legnagyobb előnye a sebesség.
- Léteznek hatékonyabb tömörítők (pl. *Monkey's Audio*) de azok lassabbak.

Neuronhálós prediktív modell

Lineáris prediktív modell

Neuronhálós prediktív modell

A tömörítés fázisai

Lineáris prediktív modell

- A leggyakrabban használt modell a **lineáris prediktív modell**.
- A lényege az, hogy a jósolt érték az előző **N** érték lineáris kombinációjával egyenlő.
- Ahhoz, hogy a jóslás pontos legyen, meg kell találni a megfelelő együtthatókat.
- Előny: gyors és hatékony az esetek többségében
- Hátrány: komplex hullámokat nem tud elég jól megjósolni a linearitása miatt

Lineáris prediktív modell

- Lineáris jóslás:

$$S_i = S_{i-1}w_1 + S_{i-2}w_2 + \dots + S_{i-N}w_N$$

- Minden előző **N** mintához (s) tartozik egy súly (**w**).
- Minden *frame* esetében a minták kódolása előtt ki kell számolni a súlyokat.
- A modell kódolása során ezt az **N** darab súlyt kell eltárolni.

Neuronhálós prediktív modell

- Ahhoz, hogy csökkenteni tudjuk a hibák mértékét, egy **nemlineáris modellt** kell alkalmaznunk.
- Ebben az esetben viszont sokkal nehezebb és erőforrásigényesebb kiszámolni a megfelelő együttthatókat.
- Erre a problémára ad megoldást a **neuronháló**, ami egy adaptív, nemlineáris függvényapproximációs rendszer. Az egyik legnagyobb hátránya, hogy relatív lassú.
- Egy ilyen neuronhálónak kell legyen **N** bemenete és egy kimenete. Több hangcsatorna esetén meg lehet többszörözni ezeknek a számát a nagyobb hatékonyság érdekében.

A tömörítés fázisai

- Egy *frame* tömörítése a következő fázisokból áll:
 - ▣ a modell létrehozása, vagyis a neuronháló betanítása
 - ▣ a modell, vagyis a kapott együtthatók (súlyok), kódolása
 - ▣ a reziduális jel kiszámolása és kódolása

- Kitömörítés:
 - ▣ a modell dekódolása
 - ▣ a reziduális jel dekódolása
 - ▣ az eredeti jel kiszámolása

Eredmények

Implementáció

A hibaértékek statisztikai eloszlása

A tömörítés mértéke N függvényében...

...és a rejtett neuronok száma függvényében

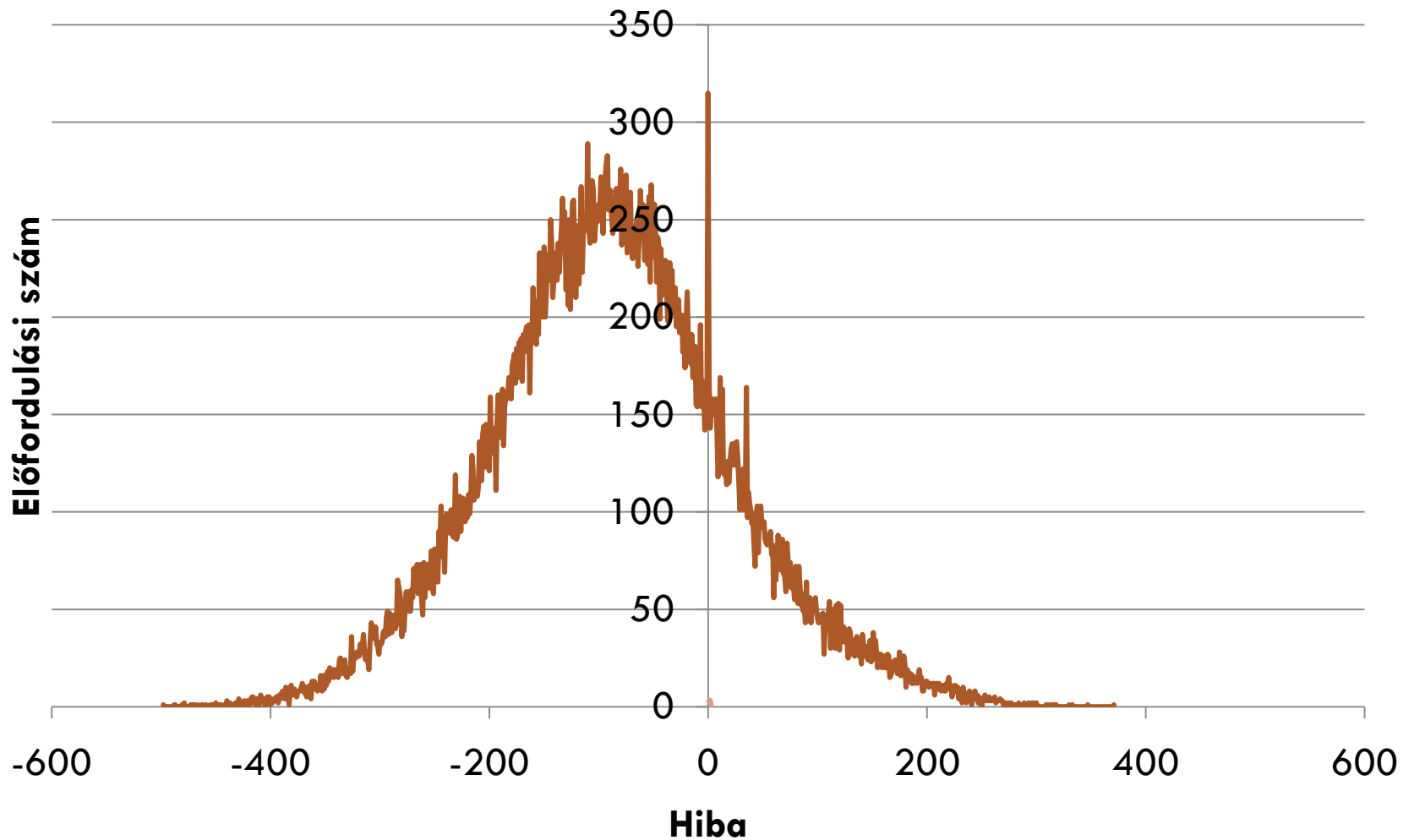
Tömörítők összehasonlítása

Következtetések

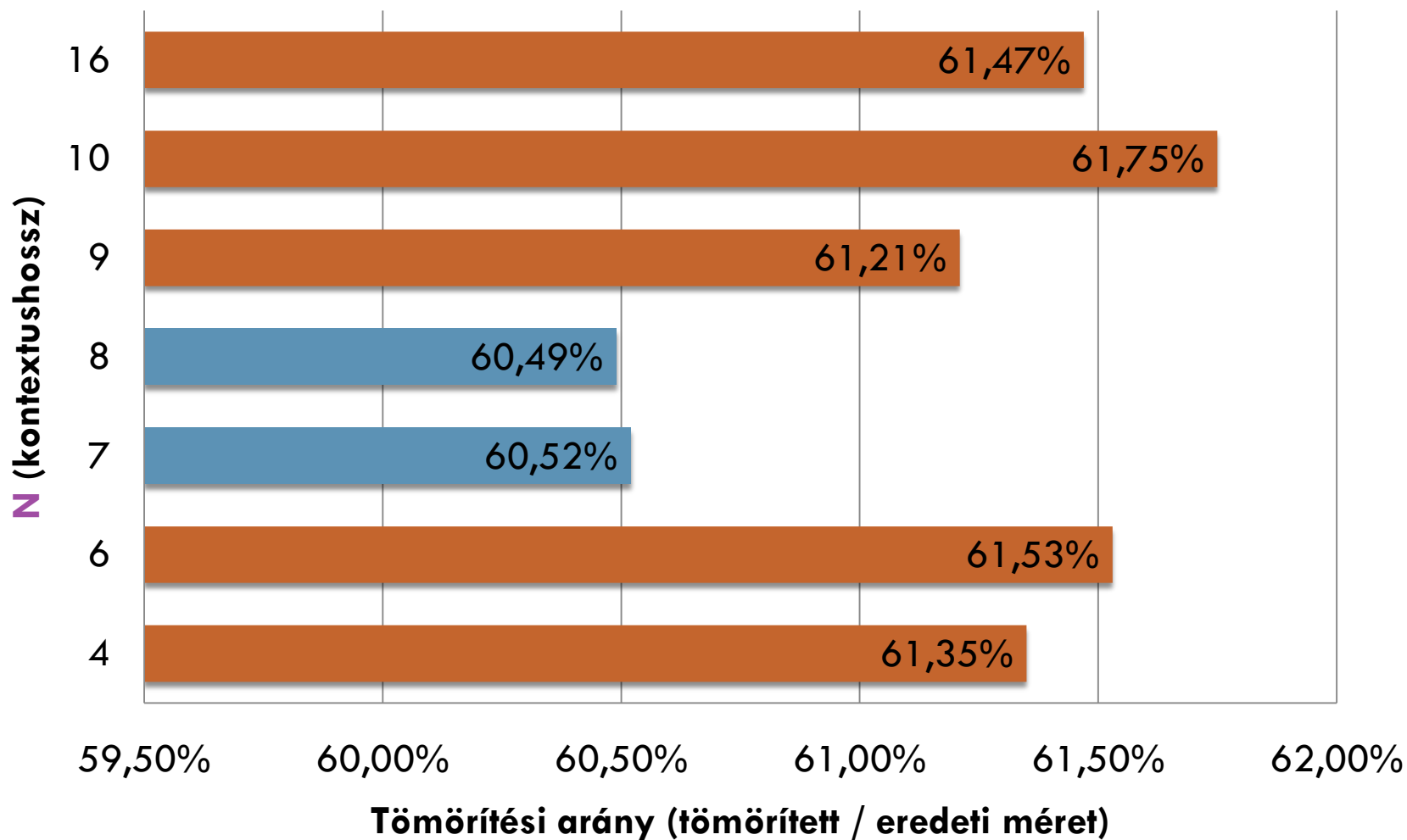
Implementáció

- Az algoritmus C++ nyelvben és a FANN neuronháló könyvtár felhasználásával lett implementálva.
- Az alkalmazott neuronháló-tanítási algoritmus: RPROP
- A program kódolás helyett kiszámolja az ideális tömörített méretet, ami aritmetikai kódolással majdnem elérhető.
- A tesztadatot egy 65536 mintából álló 16 bites pontosságú hangfájl képezte (tehát a fájl tömörítetlen mérete 128 KB).

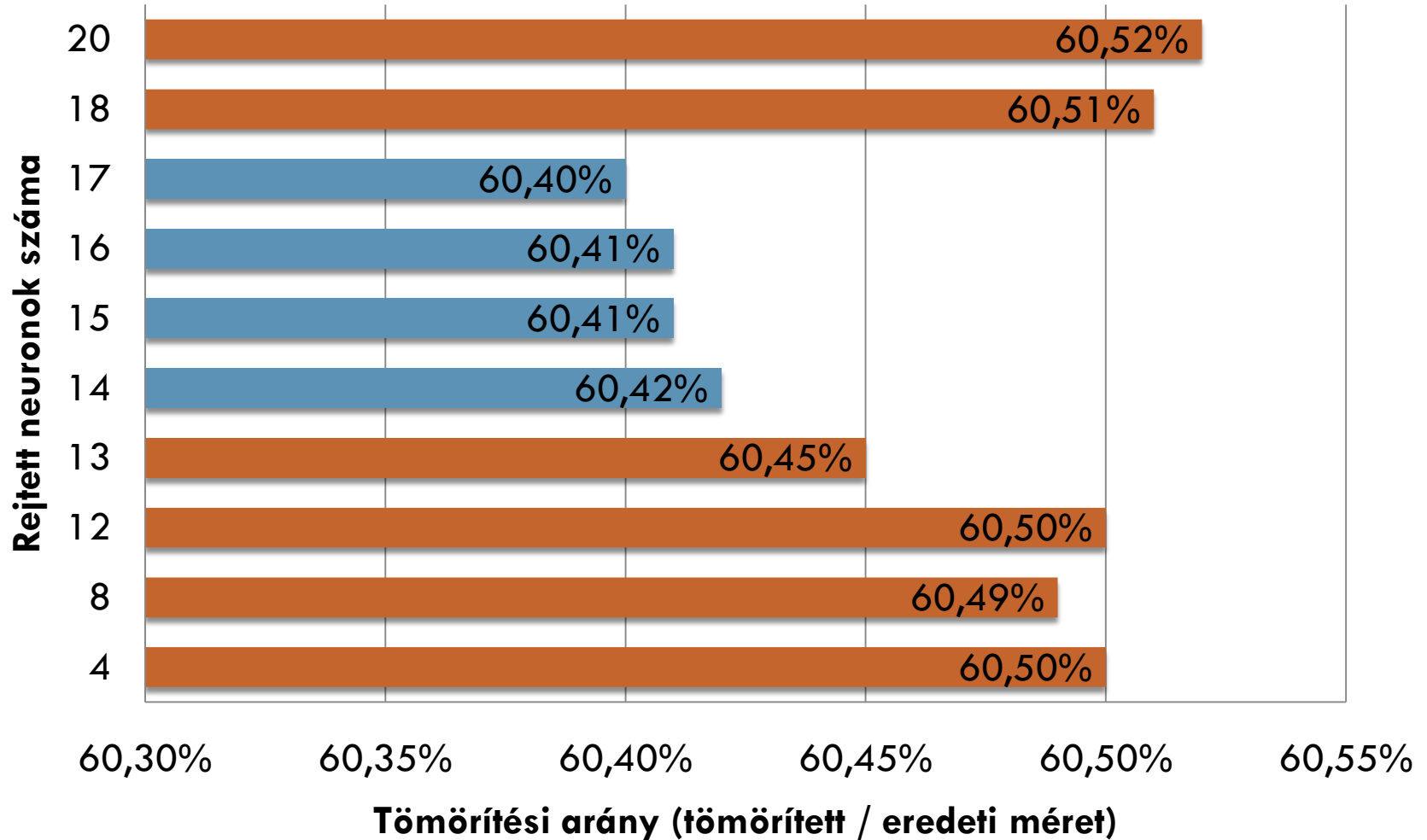
A hibaértékek statisztikai eloszlása



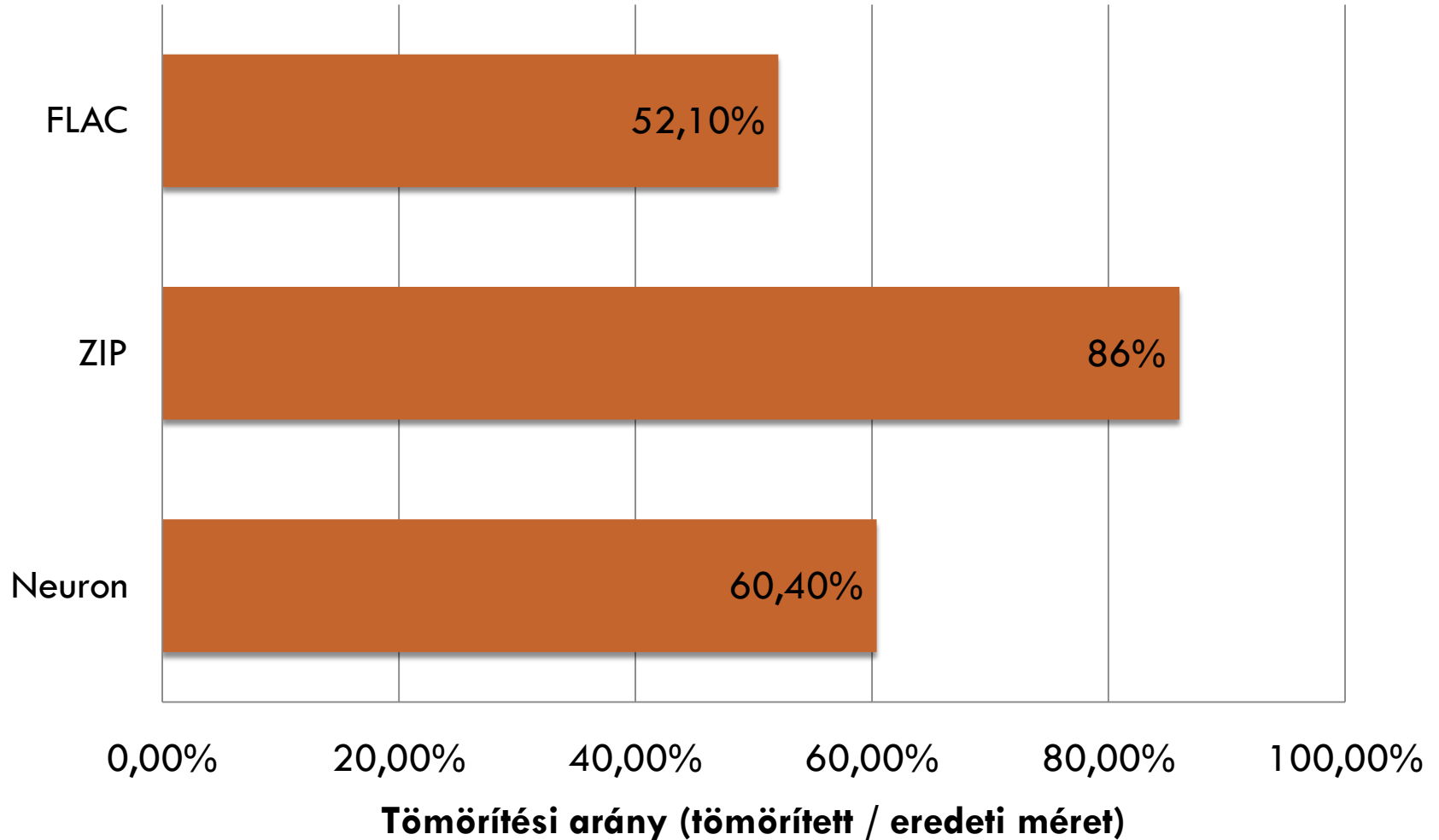
A tömörítés mértéke **N** függvényében...



...és a rejtett neuronok száma függvényében ($N = 8$)



Tömörítők összehasonlítása



Következtetések

- A tömörítés abban az esetben a legjobb, ha **N** 7 vagy 8, egyetlen rejtett réteg van és a benne levő neuronok száma 14 és 17 között van.
- Azért nem éri el a FLAC teljesítményét, mert abban az esetben a kódolás további optimalizációs algoritmusokat is tartalmaz. Ezeket a bemutatott módszerbe is be lehet építeni.

Források

<http://flac.sourceforge.net>

http://flac.sourceforge.net/documentation_format_overview.html

<http://flac.sourceforge.net/format.html>

[http://en.wikipedia.org/wiki/Pulse code modulation](http://en.wikipedia.org/wiki/Pulse_code_modulation)