Radar for Advanced Vehicular Systems

Radar Signal Processing with ANN for DOA Estimate

Microelectronics Design Automation w/ OpenSource CAD

Mahdi Haghzadeh, PhD
Cognitave Inc
http://cognitave.com ➞ tex@cognitave.com
Advanced Vehicular Systems

Autonomous Driving

Direction Finding MUSIC (Multiple Signal Classifier)
Novel Chipset (AI and or Neuromorphic)
AI Chips
  Mythic
  Intel Nervana
  IBM TrueNorth
  NXP MRAM > Neuromorphic
AI Licensing
  Brain in Silicon Stanford
  Numenta

Preliminary Matlab and Simulink Work (Simulations and Results)
Ne = 6

Ne > Number of Antenna Elements
Ne = 24

Ne > Number of Antenna Elements
Ne = 96

Ne > Number of Antenna Elements
Compare Results for Different Ne

Three moving target in from of vehicle. Variable number of radar elements.

- Ne = 6
- Ne = 24
- Ne = 96
Ne = 6

- True Rng | Est. Rng | True Sp | Est. Sp | True Ang | Est. Ang
- 50.0000  49.9586  96.0000  97.3971  45.0000  45.0001
- Error Rng | Error Sp | Error Ang
- 0.0414  -1.3971  -0.0001
Ne = 24

- True Rng | Est. Rng | True Sp | Est. Sp | True Ang | Est. Ang
- 50.0000   49.9584   96.0000   97.3973   45.0000   45.0003
- Error Rng | Error Sp | Error Ang
- 0.0416   -1.3973   -0.0003
Ne = 96

- True Rng | Est. Rng | True Sp | Est. Sp | True Ang | Est. Ang
- 50.0000 | 49.9582 | 96.0000 | 97.3822 | 45.0000 | 45.0003
- Error Rng | Error Sp | Error Ang
- 0.0418 | -1.3822 | -0.0003

Range-Angle Response Pattern

Range-Speed Response Pattern
Direction of Arrival DOA mainly is calculated using MUSIC algorithm which is computationally expensive. Conventional analytical or numerical MUSIC algorithm result in delays in RSP, heating up the chipsets and radar, low S/N. Consequently this results in diminished performance (lower range and resolution performance).

Could an Artificial Neural Network solve DOA?
DOA - ANN - Ne 24
Simulink Matlab 2018
DOA - ANN - Ne 96

Simulink Matlab 2018
Comparison of Neural Network Sims

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Data Division: Random (dividend)</th>
<th>Training: Levenberg-Marquardt (trainlm)</th>
<th>Performance: Mean Squared Error (mse)</th>
<th>Calculations: MEX</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Progress</th>
<th>Epoch:</th>
<th>0</th>
<th>386 Iterations</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time:</td>
<td>0.00-00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance:</td>
<td>1.78e+04</td>
<td>0.0994</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Gradient:</td>
<td>5.45e+04</td>
<td>0.0758</td>
<td>1.00e-07</td>
<td></td>
</tr>
<tr>
<td>Mu:</td>
<td>0.00100</td>
<td>0.0100</td>
<td>1.00e+10</td>
<td></td>
</tr>
<tr>
<td>Validation Checks:</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Data Division: Random (dividend)</th>
<th>Training: Levenberg-Marquardt (trainlm)</th>
<th>Performance: Mean Squared Error (mse)</th>
<th>Calculations: MEX</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Progress</th>
<th>Epoch:</th>
<th>0</th>
<th>570 Iterations</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time:</td>
<td>0.00-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance:</td>
<td>1.94e+04</td>
<td>0.0816</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Gradient:</td>
<td>5.64e+04</td>
<td>25.0</td>
<td>1.00e-07</td>
<td></td>
</tr>
<tr>
<td>Mu:</td>
<td>0.00100</td>
<td>0.0100</td>
<td>1.00e+10</td>
<td></td>
</tr>
<tr>
<td>Validation Checks:</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Data Division: Random (dividend)</th>
<th>Training: Levenberg-Marquardt (trainlm)</th>
<th>Performance: Mean Squared Error (mse)</th>
<th>Calculations: MEX</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Progress</th>
<th>Epoch:</th>
<th>0</th>
<th>856 Iterations</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time:</td>
<td>0.00-46</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance:</td>
<td>1.29e+04</td>
<td>0.00622</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Gradient:</td>
<td>1.07e+05</td>
<td>1.65</td>
<td>1.00e-07</td>
<td></td>
</tr>
<tr>
<td>Mu:</td>
<td>0.00100</td>
<td>0.00100</td>
<td>1.00e+10</td>
<td></td>
</tr>
<tr>
<td>Validation Checks:</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
DOA - ANN vs Root Music

Findings

• **ANN MUSIC** is more than 30 x faster for higher Ne (number of elements)

• ANN has an accuracy better than 0.1%, i.e. 0.1 degree in 100 degree.

• Root Music has 100x more accuracy, which is unnecessary.

• **ANN yields much higher CPU processing speeds.** Higher speeds and low power consumption possible with Analog / Neuromorphic RSP

• ANN benefits from mass parallelism and doesn’t slow down with Ne scale up, given than the number of ANN inputs can be reduced for higher Ne

<table>
<thead>
<tr>
<th>Ne</th>
<th>ANN CPU Time (s)</th>
<th>MUSIC CPU Time (s)</th>
<th>Processing Speed Up (times)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ne = 6</td>
<td>0 - 0.01</td>
<td>0.02 - 0.03</td>
<td>2x - Inf</td>
</tr>
<tr>
<td>Ne = 24</td>
<td>0 - 0.01</td>
<td>0.03 - 0.04</td>
<td>3x - Inf</td>
</tr>
<tr>
<td>Ne = 96</td>
<td>0.02 - 0.03</td>
<td>0.50 - 0.53</td>
<td>30x - Inf</td>
</tr>
</tbody>
</table>
Next Steps
Towards Neuromorphic Processing

- DOA w/ ANN
  - Two targets, super resolution
  - Run for Ne 24 and 96
  - Improve Accuracy
- Neuromorphic implementation
  - New memory-processor architecture
  - New data handling approach
  - CFAR
- Circuit level neuromorphic RSP
Micro-Electronics Design Automation
Open source, free of charge software

>> Radar Unit Design-flow

- Design
- Simulation & Modeling
- Fabrication & Prototyping
- RF/MW Testing
- Radar Signal Processing
- Validation Verification
- Installation
Licensed CAD Design Flow

• Licenses are expensive and accessed via network/token based
• In-house customization is possible but limited
Open-Source CAD Design Flow

- Open source, free of charge EDA is customizable & integratable
- Python based macro tool scripting possible
Open-Source CAD: Continued

Circuit and System level Analysis

LTspice®

QucsStudio

QSPICE™

1st Generation SPICE Circuit Simulation
Thank you!
Please email us with any questions.

Mahdi Haghzadeh, PhD
Cognitave Inc.
http://cognitave.com ➞ tex@cognitave.com
Artificial Intelligence for DoA Estimation

Fundamentals

1. Detect targets using RV image or whole cube (coherent or incoherent processing)
2. Get spatial information from all $M$ antennas from RV bin containing target
3. Perform DoA Estimation for each target

Equivalent complex model in time-domain

\[
\begin{bmatrix}
    b'_1[k, l] \\
    \vdots \\
    b'_M[k, l]
\end{bmatrix} = \sum_{k=0}^{K-1} \begin{bmatrix}
    a_1(\theta_k, \phi_k) \\
    \vdots \\
    a_M(\theta_k, \phi_k)
\end{bmatrix} b'_0[k, l] = \exp(j \varphi_b[k, l])
\]

Extracted Vector

\[
x \approx \sum_{k=0}^{K-1} P_K \begin{bmatrix}
    a_1(\theta_k, \phi_k) \\
    \vdots \\
    a_M(\theta_k, \phi_k)
\end{bmatrix} + n, \quad n \sim N(0, \sigma_{2D-FFT})
\]

Single-Snapshot Spatial Covariance Estimate

\[
\hat{R}_{xx} = xx^H, \quad \hat{R}_{xx}^{\text{norm}} = \hat{R}_{xx}/\sqrt{x^H x}
\]

(\cdot)^H: complex conjugate transpose

DOA - Root Music

2-D MUSIC Spatial Spectrum

- Power (dB)
- Elevation Angle (degrees)
- Azimuth Angle (degrees)

Diagram:
- Antenna data
  - $X_I$
  - $X_M$
- Correlation Matrix Calculation $R_{xx}$
- Eigen Decomposition $E_N$
- MUSIC Spectrum Calculations
- Estimation of the largest peaks $PMU(\theta)$
- Angles of Arrival $\theta_1$, $\theta_d$
Artificial Intelligence for DoA Estimation

The Input to the Artificial Neural Networks

Processing Flow

- Generate synthetic data or extract from 2D-FFTs of measurements
- Compute (normalized) covariance matrix $\hat{R}_{xx}$, eliminate power information
- Extract upper-triangular part of $\hat{R}_{xx}^{\text{norm}}$ (operation $\text{triu}(\cdot)$)
- Reshape into 20-element real-valued vector: input vector $a$ for Neural Network

Direction of Arrival (DOA) mainly is calculated using MUSIC algorithm, which is computationally expensive.

Conventional analytical or numerical MUSIC algorithm result in:
- delays in RSP
- heating up the chipsets and radar
- low S/N

Consequently, this results in diminished performance (lower range and resolution performance).

Could an Artificial Neural Network solve DOA?
DOA - ANN with Matlab

- Regression Problem
- Training: 500 Epochs
- 11550 Examples per Epoch
- SNR ~40dB